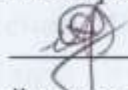


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра на тему:
«Проект програмного рішення для ідентифікації ознак
нетипової активності в блокчейн-мережах»

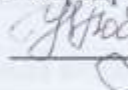
Виконала: студентка 4-го курсу групи КБ-22-1
спеціальності 125 «Кібербезпека»

 Анастасія БОДНАРАШИК

Керівник: старший викладач кафедри КІ та КБ

 Олександра ПОКОТИЛО

Рецензент: доцент кафедри ІПЗ, к.т.н., доцент

 Надія ЛОБАНЧИКОВА

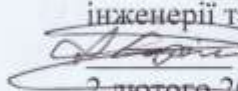
Житомир – 2026

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко

2 лютого 2026 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студентка: Боднарашик Анастасія Олександрівна

Тема роботи: «Проект програмного рішення для ідентифікації ознак нетипової активності в блокчейн-мережах»

затверджена Наказом університету від 2 лютого 2026 р. №58/с

Термін здачі студентом закінченої роботи 12 червня 2026 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): Об'єктом дослідження є процес моніторингу та аналізу транзакцій у блокчейн-мережах. Предметом дослідження є алгоритмічні методи та програмні засоби ідентифікації ознак нетипової активності у блокчейн-мережах.

Обладнання та програмне забезпечення:

1. Робоча станція з операційною системою Microsoft Windows 11
2. Середовище розробки PyCharm (v. 2026.1.2)

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Дацюк Д. В.	12.03.2026 	12.03.2026 
2	Дацюк Д. В.	15.04.2026 	15.04.2026 
3	Дацюк Д. В.	18.05.2026 	18.05.2026 

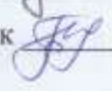
Студентка  Анастасія БОДНАРАШИК

Керівник  Олександра ПОКОТИЛО

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	12 березня 2026	Виконано
2	Розробка плану роботи	21 березня 2026	Виконано
3	Пошук інформації щодо використання криптовалют у незаконній діяльності	29 березня 2026	Виконано
4	Пошук аналогічного програмного забезпечення для перевірки криптотранзакцій на предмет підозрілості	11 квітня 2026	Виконано
5	Аналіз літературних джерел	26 квітня 2026	Виконано
6	Розробка програмного рішення	13 травня 2026	Виконано
7	Тестування програми	24 травня 2026	Виконано
8	Отримання відгуку керівника	10 червня 2026	Виконано
9	Допуск завідувача кафедри	10 червня 2026	Виконано
10	Подання роботи на рецензування	10 червня 2026	Виконано

Студентка  Анастасія БОДНАРАШИК

Керівник  Олександр ПОКОТИЛО

АНОТАЦІЯ

Кваліфікаційна робота містить опис повного циклу проєктування, розробки та налаштування автономного програмного комплексу для оперативної аналітики та первинної фільтрації транзакційних аномалій у блокчейн-мережах. Пояснювальна записка складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків. Текстова частина викладена на 50 сторінках друкованого тексту. Пояснювальна записка має 16 сторінок додатків. Список використаних джерел містить 23 найменування і займає 3 сторінки. В роботі наведено 16 рисунків. Загальний обсяг роботи – 67 сторінок.

Метою роботи є розробка та налаштування автономного програмного засобу для проактивного фінансового моніторингу та детермінованої фільтрації підозрілих блокчейн-операцій на локальній робочій станції дослідника.

У роботі проаналізовано теоретичні та правові основи функціонування ринку віртуальних активів в Україні, порівняно варіанти реалізації існуючих комерційних платформ блокчейн-аналітики та виявлено їхні архітектурні обмеження. Спроектовано математичні моделі та визначено числові й топологічні критерії нетиповості транзакцій для моделей виявлення «micro» та «whale». Здійснено обґрунтований вибір програмного стеку на базі мови Python, бібліотеки Pandas, мікрофреймворка Flask та вбудованої СУБД SQLite. Описано розробку клієнт-серверної архітектури додатка із системою захисту інформації, що включає криптографічне хешування паролів та ведення передагового журналу аудиту безпеки. Наведено результати функціонування та швидкісного тестування алгоритмів класифікації на реальному ончейн-масиві даних, а також продемонстровано аналіз виявлених патернів автоматизованої діяльності фінансових установ.

Ключові слова: БЛОКЧЕЙН, ТРАНЗАКЦІЯ, АНОМАЛІЯ, ФІЛЬТРАЦІЯ, ЕВРИСТИЧНИЙ АНАЛІЗ, PANDAS, FLASK, ЛОКАЛЬНА БАЗА ДАНИХ, КІБЕР-БЕЗПЕКА.

						ККІК.КР.Б – 125 – 02 – ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата	Проект програмного рішення для ідентифікації ознак нетипової активності в блокчейн-мережах	Літ.	Арк.	Аркушів
Розроб.		Боднаршин А.О.		11.06.24				
Керівник		Покотило О.А.		11.06.24			4	67
Реценз.		Лобанчикова Н.М.		11.06.24		ФІКТ, Гр. КБ-22-1		
Н. контр.		Сфіменко А.А.		11.06.24				
Затверд.		Сфіменко А.А.		11.06.24				

ANNOTATION

This qualification paper describes the full cycle of design, development and configuration of a standalone software suite for real-time analytics and initial filtering of transactional anomalies in blockchain networks. The explanatory note comprises an introduction, three chapters, conclusions, a bibliography and appendices. The main text comprises 50 pages of printed text. The explanatory note includes 16 pages of appendices. The list of references contains 23 entries and spans 3 pages. The thesis includes 16 figures. The total length of the thesis is 67 pages.

The aim of the work is to develop and configure a standalone software tool for proactive financial monitoring and deterministic filtering of suspicious blockchain transactions on the researcher's local workstation.

The paper analyses the theoretical and legal foundations of the virtual asset market in Ukraine, compares the implementation options of existing commercial blockchain analytics platforms, and identifies their architectural limitations. Mathematical models have been designed, and numerical and topological criteria for transaction atypicality have been defined for the 'micro' and 'whale' detection models. A well-founded choice of software stack has been made, based on the Python language, the Pandas library, the Flask microframework, and the embedded SQLite DBMS. The development of a client-server application architecture with an information security system is described, which includes cryptographic password hashing and the maintenance of an uneditable security audit log. The results of the performance and speed testing of classification algorithms on a real on-chain dataset are presented, and an analysis of the identified patterns of automated activity within financial institutions is demonstrated.

Keywords: BLOCKCHAIN, TRANSACTION, ANOMALY, FILTERING, HEURISTIC ANALYSIS, PANDAS, FLASK, LOCAL DATABASE, CYBERSECURITY.

		БоднарашикА.О.			ККИКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ ТА ВИМОГИ ДО ПРОГРАМНОГО РІШЕННЯ	10
1.1 Специфіка відмивання коштів через криптовалюти та проблема ручного моніторингу	10
1.2 Існуючі системи аналізу транзакцій та їхні обмеження	14
1.3 Визначення критеріїв нетиповості для алгоритмічного аналізу	18
1.4 Обґрунтування вибору програмного стеку	21
РОЗДІЛ 2 АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ	24
2.1 Загальна логіка роботи програми та сценарії використання	24
2.2 Розробка алгоритмів фільтрації блокчейн-даних	28
2.3 Проєктування локальної бази даних та захист інформації.....	30
2.4 Програмна реалізація бекенду та користувацького інтерфейсу	32
РОЗДІЛ 3 ТЕСТУВАННЯ ТА ПРАКТИЧНЕ ВИКОРИСТАННЯ ПРОГРАМНОГО РІШЕННЯ	35
3.1 Порядок запуску та налаштування локального середовища	35
3.2 Тестування алгоритмів класифікації на реальних даних.....	37
3.3 Аналіз результатів	41
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	48
ДОДАТКИ	51

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ВА – віртуальний актив

ПЗ – програмний засіб (або програмне забезпечення)

СУБД – система керування базами даних

API – інструментарій прикладного програмування (Application Programming Interface)

DeFi – децентралізовані фінансові сервіси (Decentralized Finance)

HTML – мова розмітки гіпертексту (HyperText Markup Language)

MVC – архітектурний шаблон проектування (Model-View-Controller)

OSINT – розвідка на основі відкритих джерел (Open Source Intelligence)

SaaS – програмне забезпечення як послуга (Software as a Service)

SQL – мова структурованих запитів (Structured Query Language)

TSV – формат текстових файлів із розділювачами-табуляціями (Tab-Separated Values)

USD – долар США (фіатна валюта)

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми. Значна частина коштів, отриманих злочинним шляхом, проходить крізь міксери та транзитні вузли перш ніж злочинці дозволять собі вільно їх витратити. Псевдоанонімність та децентралізація у блокчейн-мережах приваблює користувачів як законослухняних, так і тих, які мають на меті приховати свою особистість від правоохоронних органів задля уникнення відповідальності за скоєні правопорушення. Кожен блок містить дані про тисячі транзакцій, тому переглядати кожну на предмет підозрілості може потребувати значних трудових витрат та часу фахівців. Коли працівник дійде до підозрілої транзакції, великий шанс, що станом на той момент ідентифікувати або затримати порушника вже не вдасться. Тому має сенс проводити подібний моніторинг автоматизовано, а детальніше розглядати лише ті з транзакцій, які мають нетипові характеристики.

Метою кваліфікаційної роботи є розробка та налаштування автономного програмного засобу для проактивного фінансового моніторингу та детермінованої фільтрації транзакцій з ознаками нетипової активності у блокчейн-мережах. Використання автоматизованих засобів аналізу дозволяє суттєво скоротити обсяг транзакцій, які потребують ручної перевірки, та зосередити увагу фахівців на операціях, що мають ознаки нетипової активності.

Визначена мета обумовлює наступні завдання:

- проаналізувати особливості використання блокчейн-технологій у контексті фінансового моніторингу та кібербезпеки;
- дослідити існуючі програмні засоби аналізу блокчейн-транзакцій та їхні обмеження;
- визначити критерії та ознаки нетипової активності в блокчейн-мережах;
- розробити алгоритми виявлення транзакцій з ознаками нетипової активності;
- реалізувати програмний засіб для автоматизованого аналізу блокчейн-транзакцій;
- провести тестування розробленого програмного рішення на реальних даних.

		БоднарашикА.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Об’єктом дослідження є процес моніторингу та аналізу транзакцій у блокчейн-мережах.

Предметом дослідження є алгоритмічні методи та програмні засоби ідентифікації ознак нетипової активності у блокчейн-мережах.

У процесі виконання кваліфікаційної роботи було розроблено програмне рішення для автоматизованого виявлення транзакцій з ознаками нетипової активності у блокчейн-мережах. Розроблений програмний засіб може використовуватися як інструмент первинного відбору потенційно підозрілих транзакцій для подальшого аналізу фахівцями з кібербезпеки та фінансового моніторингу.

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ ТА ВИМОГИ ДО ПРОГРАМНОГО РІШЕННЯ

1.1 Специфіка відмивання коштів через криптовалюти та проблема ручного моніторингу

Сучасний фінансовий ландшафт зазнає безпрецедентних трансформацій під впливом технологій розподілених реєстрів, які стали фундаментом для появи віртуальних активів. Це явище виходить далеко за межі простих цифрових фінансових інструментів, створюючи нову екосистему правовідносин, що вимагає ретельного теоретичного осмислення та належного регуляторного забезпечення. Віртуальний актив сьогодні розглядається як нематеріальне благо [1], що володіє внутрішньою вартістю та виражене сукупністю електронних даних, які забезпечують його оборотоздатність у рамках відповідної системи.

Правова природа віртуальних активів в Україні закріплена як об'єкт цивільних прав, що вимагає належного регулювання [1, 4, 7]. Взаємодія з такими активами здійснюється через спеціалізоване програмне забезпечення – гаманці. Гаманець виступає технічним інтерфейсом, який за допомогою алгоритмів криптографічних ключів надає користувачеві доступ до управління належними йому віртуальними активами.

Фундаментальний поділ віртуальних активів на забезпечені та незабезпечені [1] є критично важливим для розуміння ринкових механізмів. Забезпечені активи посвідчують певні майнові права чи права вимоги на інші об'єкти цивільних прав, що робить їх більш передбачуваними та менш волатильними, на відміну від незабезпечених активів, які, фактично, не гарантують жодних прав, окрім права володіння ними. Ця класифікація лежить в основі класичного підходу до оцінки ризиків на ринку, де забезпечення виступає своєрідним якорем для інвестора.

Система забезпечення обороту віртуальних активів виконує роль технологічного каркаса, в якому фіксуються всі дії учасників: від створення активу до його виведення з цивільного обороту. Момент створення віртуального активу визначається точкою, з якої перший власник отримує можливість повноцінного контролю,

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

що часто корелює з технічними особливостями обраного блокчейну. Така прозорість, хоч і здається абсолютною, насправді створює складний простір для аналітичних методів дослідження транзакцій.

Ринок віртуальних активів об'єднує постачальників послуг та фізичних чи юридичних осіб, що діють у власних інтересах, формуючи мережу правовідносин між ними та державою. Постачальники послуг, як-от майданчики з обміну чи сервіси зберігання, виступають ключовими вузлами цієї екосистеми, на яких покладаються обов'язки з дотримання вимог фінансового моніторингу. Це особливо актуально в контексті боротьби з легалізацією доходів, отриманих злочинним шляхом, де діяльність учасників ринку повинна відповідати міжнародним та національним стандартам безпеки.

Право власності на віртуальний актив у цій новій парадигмі засвідчується володінням відповідним ключем, що переводить питання права власності у площину технічного контролю. Алгоритми, що визначають умови набуття та переходу прав, часто жорстко закодовані в самому реєстрі, що мінімізує вплив людського фактора, але одночасно створює нові виклики для захисту прав власника у разі неправомірного заволодіння таким ключем.

Національне законодавство чітко розмежовує фінансові віртуальні активи, забезпечені валютними цінностями або цінними паперами, від інших видів, встановлюючи для них різні рівні державного контролю [2, 3]. Така сегментація є стратегічною для держави, адже дозволяє інтегрувати віртуальні активи у загальну фінансову систему, мінімізуючи ризики для стабільності національної валюти. Віртуальні активи, своєю чергою, не розглядаються як засіб платежу, що підкреслює їх статус як об'єкта інвестування, а не повноцінної заміни грошовій одиниці [5].

У контексті аналізу ринку, метод кластеризації транзакцій стає невід'ємним інструментом для виявлення підозрілих схем. Оскільки учасники часто використовують множинні адреси для приховування реальних обсягів своїх операцій, методи кластеризації дозволяють ефективно об'єднувати ці адреси в єдині логічні сутності. Це дає змогу Compliance-офіцерам та аналітикам відстежувати рух коштів через

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

довгі ланцюжки транзакцій, виявляючи патерни, що вказують на спроби відми-
вання грошей.

Сучасні підходи до моніторингу потребують глибокої експертизи, оскільки
злочинці – як свідчить історія функціонування маркетплейсів даркнету типу Silk
Road – постійно адаптують свої схеми [8, 10, 17, 20]. Кіберполіція та інші контро-
люючі органи активно використовують програмні рішення для з’єднання окремих
транзакцій у цілісні «ланцюжки» [6, 5], що дозволяє бачити повну картину руху
віртуальних активів і локалізувати точки виходу в фіатну систему.

Регуляторне середовище також вимагає від постачальників послуг впрова-
дження процедур суворої автентифікації та захисту персональних даних. Питання
кібербезпеки стають не просто внутрішньою справою компаній, а ліцензійною ви-
могою, адже слабкість інформаційних систем постачальника послуг ставить під за-
грозу активи тисяч користувачів. Реєстрація постачальників у спеціалізованих дер-
жавних реєстрах є одним із кроків до створення прозорого та підзвітного ринку.

Координація діяльності між регуляторами, такими як Національна комісія з
цінних паперів та фондового ринку та Національний банк України, створює умови
для ефективного нагляду. Вони обмінюються інформацією про банки, небанківські
установи та інші суб’єкти, що створює єдиний інформаційний простір для боротьби
зі зловживаннями. Державний контроль поширюється не лише на самих постача-
льників послуг, а й на емітентів віртуальних активів, висуваючи вимоги до проспек-
тів емісії та розкриття публічної інформації.

Використання фінансових санкцій за порушення законодавства про віртуа-
льні активи є дієвим запобіжником проти безконтрольної діяльності неавторизова-
них суб’єктів. Відсутність дозволів на надання певних послуг, надання недостовір-
ної інформації чи порушення строків подання звітності призводять до значних
штрафів, що стимулює постачальників послуг дотримуватися встановлених пра-
вил.

Міжнародне співробітництво у цій сфері стає критично важливим, оскільки
віртуальні активи за своєю природою не мають кордонів. Імплементация стандартів

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

FATF є обов'язковою умовою для інтеграції українського ринку у світовий, що дозволяє отримувати доступ до міжнародних баз даних та досвіду інших країн у боротьбі з кримінальною діяльністю в криптосфері. Принципи адекватності, ефективності та передбачуваності в регуляторній діяльності допомагають підтримувати баланс між розвитком інновацій та безпекою системи.

Окремим викликом є боротьба з використанням номінальних рахунків, де злочинці експлуатують вразливі категорії громадян для проведення сумнівних операцій. Такі схеми створюють значне навантаження на банківську систему та регулятора, вимагаючи постійного вдосконалення алгоритмів платіжного моніторингу. Банки змушені закривати десятки тисяч таких рахунків, але злочинці знову знаходять нові способи залучення людей, що підкреслює необхідність не лише технічних, а й просвітницьких заходів.

Роль технологічного оператора платіжних послуг у цьому контексті полягає в забезпеченні надійності обробки інформації та захисту транзакційних даних. Ці суб'єкти не завжди мають прямий доступ до коштів користувачів, але їхня роль у процесингу та клірингу робить їх критично важливими компонентами інфраструктури, що вимагає відповідного рівня захисту від кіберзагроз.

Процеси автентифікації, включаючи посилену автентифікацію, стають обов'язковим стандартом безпеки для запобігання несанкціонованому доступу до активів користувачів. Використання багатофакторних методів підтвердження особи, що базуються на поєднанні різних категорій даних – знання, володіння, приналежності – суттєво знижує ймовірність успішних шахрайських атак у дистанційних платіжних операціях.

Електронна комерція в Україні, регульована окремим законом, тісно переплітається з ринком віртуальних активів, оскільки обидві сфери використовують схожі принципи вчинення дистанційних правочинів. Юридична рівність електронних договорів з письмовими формами забезпечує стабільність цивільного обороту, створюючи надійну правову базу для розвитку цифрових послуг.

Взаємодія між суб'єктами платіжних систем та операторами інфраструктури регулюється як правилами самих систем, так і загальними вимогами законодавства.

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Це забезпечує безперервність, надійність та ефективність виконання фінансових операцій, що є основою довіри до всієї фінансової системи країни. Оверсайт платіжної інфраструктури виступає тим механізмом, який контролює ці процеси на державному рівні.

Важливість збереження даних про транзакції протягом тривалого часу, принаймні протягом строків позовної давності, є архітектурним компонентом безпеки. Це дозволяє здійснювати ретроспективний аналіз у разі виникнення підозр щодо легальності минулих операцій, що є необхідним у процесі судового розгляду справ про правопорушення у цій сфері.

Підсумовуючи, можна стверджувати, що ринок віртуальних активів в Україні переходить від стадії формування до фази зрілого державного регулювання. Синергія технічних інструментів кластеризації, законодавчих норм щодо верифікації учасників та суворих вимог до безпеки інформаційних систем створює підґрунтя для безпечного та прозорого обігу віртуальних активів у майбутньому.

Незважаючи на значну кількість наявних механізмів державного регулювання та фінансового моніторингу, аналіз блокчейн-транзакцій залишається складним завданням через великі обсяги даних та псевдоанонімний характер мереж. У зв'язку з цим актуальним є використання автоматизованих програмних засобів, здатних здійснювати первинний відбір транзакцій з ознаками нетипової активності для подальшого аналізу фахівцями з кібербезпеки та фінансового моніторингу.

1.2 Існуючі системи аналізу транзакцій та їхні обмеження

Актуальність розробки інструментів аналізу підтверджується періодичною активізацією «сплячих» гаманців, пов'язаних із відомими кримінальними майданчиками минулого, що вимагає постійного моніторингу [15, 16].

Першим важливим інструментом для аналізу є платформа Chainalysis, що спеціалізується на візуалізації графів транзакцій та автоматизованій атрибуції гаманців. Її функціонал базується на велетенських базах даних, які пов'язують анонімні криптографічні адреси з реальними суб'єктами, такими як біржі, майнінг-

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

пули або відомі даркнет-ринки [12, 14]. Цей інструмент дозволяє аналітику не просто бачити переказ коштів, а відстежувати їхній шлях через десятки проміжних вузлів у зручному візуальному форматі. Проте, така система є хмарним SaaS-рішенням, яке потребує передплати та постійного доступу до інтернету, що робить її недоступною для локальної роботи з власними файлами даних без передачі інформації третім особам.

Другим значущим аналітичним продуктом є Elliptic – сервіс автоматизованого оцінювання ризиків. Він діє за принципом надання API-інтерфейсу, який в режимі реального часу присвоює кожній транзакції відповідний бал ризику [13]. Алгоритм системи автоматично порівнює характеристики операції з великою базою індикаторів компрометації. Якщо транзакція взаємодіє з адресами, які раніше були помічені у хакерських атаках чи шахрайстві, система миттєво позначає її як підозрілу [13]. Це потужний інструмент для фінансових організацій, але він не надає користувачу гнучкості у налаштуванні власних фільтрів, обмежуючись лише тими критеріями, які заклали розробники сервісу.

Третім інструментом є платформа Nansen, що спеціалізується на аналітиці ланцюжкових даних, зокрема на кластеризації гаманців та моніторингу активності великих гравців ринку [21]. Система дозволяє бачити, які активи накопичуються на певних гаманцях та куди вони переміщуються [21]. Функціонал цього інструменту є надзвичайно глибоким, але він орієнтований на дослідження ринкових трендів, а не на криміналістичний аналіз дрібних нетипових транзакцій, які є предметом дослідження даної роботи.

Окрему категорію складають загальнодоступні провідники блокчейну, такі як Etherscan або Blockchain.com, які використовує переважна більшість аналітиків-початківців [9, 19]. Вони дозволяють переглянути вміст конкретного блоку або баланс конкретної адреси. Однак ці інструменти взагалі не мають функцій автоматизованого аналізу. Для отримання інформації про великі обсяги транзакцій досліднику довелося б вручну переглядати кожну сторінку, що фізично нівелює можливість вчасного виявлення злочинної активності. Вони є надійним джерелом сирих даних, але не є інструментом для їх обробки.

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

Розроблюваний програмний засіб суттєво відрізняється від зазначених рішень за моделлю експлуатації та цільовим призначенням. Більшість комерційних платформ (Chainalysis, Elliptic, Coinbase Tracer) є закритими системами з високою вартістю ліцензування. Вони працюють у хмарі, що автоматично створює ризики конфіденційності під час завантаження даних розслідування на сторонні сервери.

Автономне програмне рішення працює виключно на локальній станції дослідника. Це гарантує повну ізоляцію даних від зовнішніх мереж. У контексті кібербезпеки це є важливою перевагою, оскільки результати аналізу, які можуть містити чутливу інформацію про хід розслідування, не потрапляють на сервери вендорів.

Також суттєвою відмінністю є можливість налаштування параметрів вибірки. У великих корпоративних системах критерії ризику часто є «чорною скринькою». Користувач отримує готовий бал ризику, але не завжди розуміє, за якими саме параметрами він був розрахований. Реалізований підхід дозволяє чітко задати математичні фільтри: обсяг транзакції, кількість входів, розмір комісії.

Ще одним аспектом є швидкість обробки. SaaS-інструменти часто обмежені пропускнуою здатністю API або квотами на кількість запитів. Алгоритмічне забезпечення обробляє стільки даних, скільки дозволяє потужність локального процесора та оперативна пам'ять комп'ютера. Це робить його ефективним для роботи з великими локальними архівами файлів .tsv.

Важливо наголосити, що даний інструментарій не претендує на роль повноцінної заміни глобальним аналітичним платформам з точки зору масштабу даних. Глобальні системи індексують весь блокчейн за десятиліття. Даний інструмент орієнтований на первинне сортування конкретного масиву, який аналітик вже отримав для дослідження.

Це дозволяє використовувати запропонований метод як «робочу конячку» на початковому етапі розслідування. Він відсіює основний обсяг легітимних операцій за короткий проміжок часу. Тільки після звуження кола підозр за допомогою такого інструменту можна переходити до використання складніших систем для детального графового аналізу конкретних адрес [13, 21].

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Унікальність архітектури полягає у простоті. Вона не вимагає складної інфраструктури, розгортання серверів чи налаштування хмарних баз даних. Програма працює за принципом прямого доступу до файлової системи, що робить її мобільною та придатною для використання в умовах обмеженого технічного забезпечення.

Використання мови Python та бібліотеки Pandas дозволяє забезпечити високу швидкість обробки табличних даних. Це технічне рішення є оптимальним для задач такого класу, де важлива не швидкість доступу до мережі, а швидкість математичних обчислень над локально доступними масивами.

Система авторизації в розроблюваній системі додає ще один рівень безпеки, якого часто не вистачає простим локальним скриптам. Захист бази даних паролем забезпечує цілісність накопиченої аналітичної інформації. Навіть у випадку несанкціонованого доступу до робочої станції, дані розслідувань залишаються захищеними.

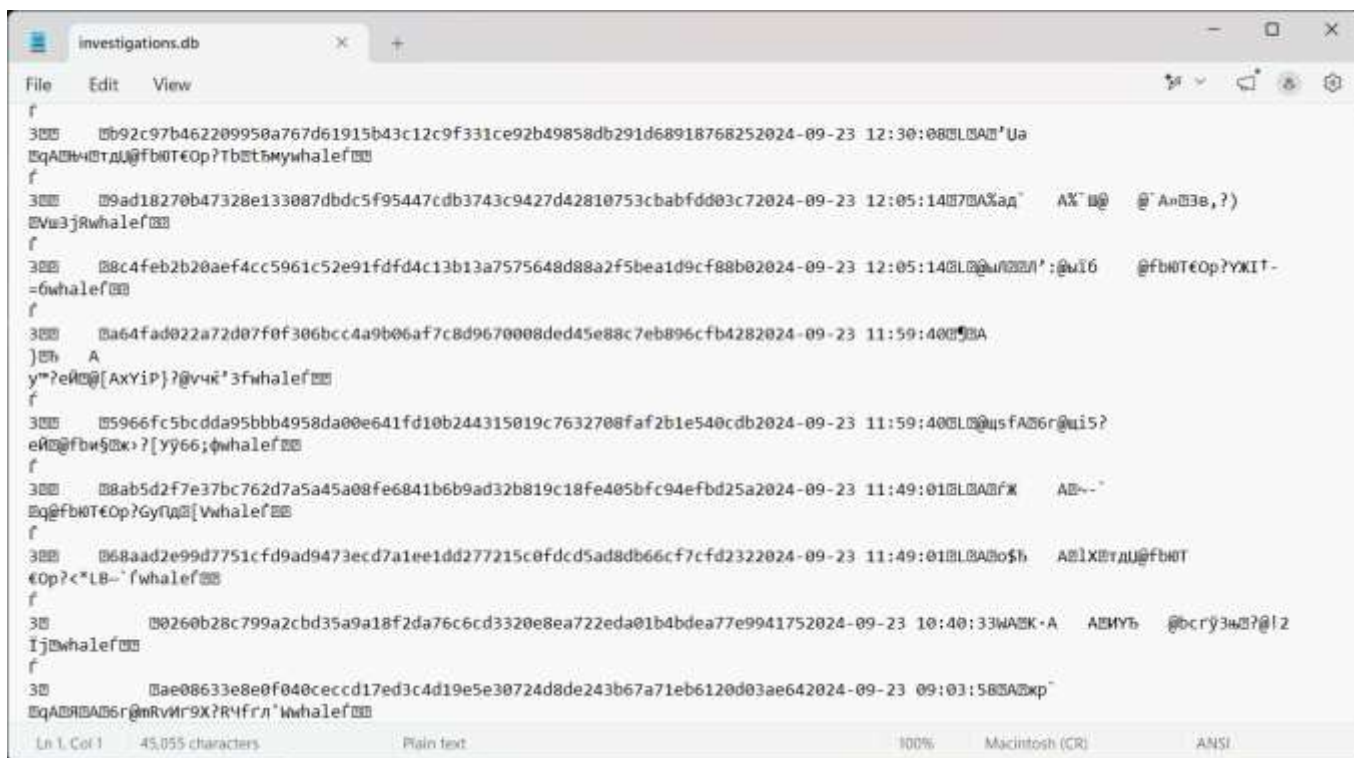


Рис.1.1 – Вміст захищеної бази даних

		Боднаришук А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

Отже, на відміну від корпоративних аналітичних платформ, розроблюваний засіб фокусується на автономності та контролі. Він не намагається охопити всі можливі вектори загроз одночасно. Замість цього, він надає інструмент для швидкого, контрольованого та прозорого аналізу даних без залежності від хмарних сервісів. Такий підхід є більш доцільним для задач оперативного розслідування, де аналітику важливо мати інструмент, який працює автономно. Він доповнює великі аналітичні системи, а не намагається їх замінити, виконуючи функцію фільтра першого рівня для попереднього відбору транзакцій, що потребують додаткового аналізу.

1.3 Визначення критеріїв нетиповості для алгоритмічного аналізу

Процес алгоритмічного аналізу блокчейн-даних потребує чіткого математичного розмежування між легітимною фінансовою активністю та операціями з ознаками приховування коштів. Оскільки мережі функціонують на принципах псевдоанонімності, ідентифікація ризиків базується на кількісних показниках кожної транзакції. Розробка автоматизованої системи моніторингу вимагає формування строгих критеріїв нетиповості, які відкидають суб'єктивні оцінки та спираються виключно на числові значення масиву даних.

Першим фундаментальним критерієм для фільтрації є загальний обсяг транзакції в доларовому еквіваленті. Використання фіатної оцінки дозволяє нівелювати волатильність курсу криптовалют та оцінювати реальну економічну вартість переказу. Алгоритм розділяє масив даних на сегменти, відсіюючи стандартні користувачькі перекази та фокусуючись на специфічних цільових групах, таких як схеми мікро-відмивання та великі інституційні переміщення.

Для виявлення схем послідовного відмивання коштів або використання алгоритмічних ботів встановлюється жорсткий коридор обсягів. Транзакції, сума яких становить від 1 000 до 20 000 доларів США, утворюють першу групу ризику. Цей діапазон відфільтровує інформаційний шум від щоденних роздрібних платежів, але залишається нижче порогу, який традиційно привертає увагу систем макро-моніторингу [14].

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

Другим полюсом аналізу є моніторинг активності інституційних суб'єктів або так званих «китів». Критерій нетиповості тут базується на фіксації масивних обсягів капіталу (транзакцій, що перевищують 100 000 доларів США). Переміщення таких сум вимагає окремого рівня контролю, оскільки вони здатні забезпечити швидке виведення значних кримінальних активів в один клік [12].

Окрім фінансового обсягу, ключовим параметром є топологія транзакції, що виражається у кількості вхідних та вихідних адрес. Легітимні комерційні сервіси та біржі зазвичай генерують транзакції з великою кількістю виходів, об'єднуючи платежі для економії комісій. Обмеження кількості входів та виходів стає критичним фільтром для відсіювання такого корпоративного трафіку [21].

У алгоритмічній моделі для виявлення «мікро-відмивання» застосовується жорстке топологічне обмеження (кількість входів та виходів не повинна перевищувати 2). Ця математична умова ідеально описує структуру класичного ланцюжка відмивання, коли один вхід розщеплюється на один цільовий переказ та одну транзакцію решти, яка повертається власнику для подальшого дроблення [13, 21].

Для виявлення великих підозрілих операцій топологічний критерій адаптується. Алгоритм обмежує кількість виходів до 5. Ця межа є статистично обґрунтованою: вона дозволяє ізолювати індивідуальні масштабні перекази від рутинних пакетних розсилок криптовалютних бірж. Транзакція на сотні тисяч доларів із мінімальною кількістю отримувачів має значно вищий профіль ризику.

Розмір сплаченої комісії за обробку транзакції виступає третім важливим індикатором поведінки ініціатора. У стандартних умовах користувачі оптимізують витрати, обираючи мінімально необхідну комісію для підтвердження. Нетипово висока комісія свідчить про штучне стимулювання майнерів з метою максимально швидкого включення транзакції у блок.

Для транзакцій середнього обсягу (від 1 000 до 20 000 доларів США) маркером аномалії є абсолютна сума комісії, що перевищує 50 доларів. У цьому фінансовому діапазоні такі витрати є економічно нераціональними для звичайного користувача. Проте для суб'єкта, який намагається швидко розірвати ланцюжок відстеження, швидкість переказу превалює над його вартістю [14].

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Для масивних переказів (понад 100 000 доларів) поріг підозрілості комісії було встановлено на рівні 100 доларів США. Плата таких сум за одну операцію вказує на критичну необхідність гарантованого та швидкого проведення коштів, що часто супроводжує процеси ліквідації активів, отриманих сумнівним шляхом [12, 13].

Додатковим розрахунковим параметром, який формується у процесі алгоритмічної обробки, є відносний відсоток комісії. Він розраховується як відношення суми сплаченої комісії до загального доларового обсягу входу. Збереження цього розрахункового показника у базі даних дозволяє аналітикам згодом виявляти операції, де комісія становить нетипово велику частку від переказу.

Формування комплексних евристичних моделей відбувається шляхом одночасного застосування всіх описаних критеріїв. Жоден із цих параметрів не є достатнім для прийняття рішення самостійно. Перетин цих умов створює вузький математичний коридор, у який потрапляють операції з найвищим ступенем ризику.

Модель виявлення мікро-відмивання спрацьовує виключно тоді, коли обсяг знаходиться в заданому середньому коридорі, кількість входів і виходів мінімальна, а комісія перевищує встановлений ліміт. Такий жорсткий алгоритмічний відбір мінімізує кількість хибних спрацювань та економить час аналітика при роботі з табличними масивами даних.

Аналогічно, модель пошуку великих інституційних гравців активується при одночасному збігу величезного обсягу, обмеженої кількості отримувачів та завищеної абсолютної комісії. Це дозволяє ефективно розділяти сирі табличні дані на конкретні категорії загроз, які вимагають різних методів подальшого розслідування.

Варто підкреслити, що всі порогові значення, фінансові ліміти та топологічні межі фільтрації для обох моделей були визначені експериментальним шляхом на основі попереднього аналізу тестових масивів транзакцій. У процесі налаштування алгоритмів ці параметри неодноразово коригувалися на основі аналізу результатів тестування для зменшення кількості хибнопозитивних спрацювань.

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

Використання детермінованих математичних фільтрів має значну перевагу над закритими індексами комерційних алгоритмів. Дослідник точно розуміє природу кожної відібраної транзакції. Такий підхід забезпечує високу швидкість обробки великих наборів даних та робить алгоритмічний аналіз надійним інструментом первинного контролю.

1.4 Обґрунтування вибору програмного стеку

Реалізація автономного програмного комплексу для первинного аналізу та фільтрації блокчейн-транзакцій вимагає підбору технологічного стеку, здатного забезпечити високу швидкість математичних обчислень над локальними масивами даних, надійність збереження інформації та безпеку автентифікації користувачів. Оскільки розроблюване рішення орієнтоване на роботу безпосередньо на локальній робочій станції дослідника без залучення хмарних обчислювальних потужностей, ключовими критеріями вибору стали продуктивність обробки табличних файлів великого обсягу, гнучкість розробки алгоритмів та мінімальні вимоги до системних ресурсів.

Основою програмного рішення обрано мову програмування Python. Головною перевагою цього вибору є розвинена екосистема спеціалізованих бібліотек для аналізу даних та наявність високорівневих фреймворків для створення архітектури вебдодатків. У порівнянні з компільованими мовами, такими як C# або Java, Python дозволяє значно швидше проектувати та модифікувати математичні фільтри евристичного аналізу. Недоліком Python є нижча швидкість виконання чистих циклів, проте цей мінус повністю нівелюється інтеграцією низькорівневих оптимізованих бібліотек для аналізу даних.

Для маніпуляцій із масивами даних формату .tsv (значення, розділені табуляцією) обґрунтовано використано бібліотеку Pandas. Альтернативою розглядалося застосування вбудованого модуля Python csv або розробка власних парсерів файлів. Проте класичний пооб'єктний перебір рядків через вбудований модуль csv демонструє низьку швидкість обробки великих архівів даних і потребує значних обсягів

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

написання коду для фільтрації. Бібліотека Pandas використовує векторизовані операції, реалізовані мовами C та Cython, що забезпечує виконання математичних фільтрацій над сотнями тисяч рядків транзакцій за мілісекунди, оптимізуючи навантаження на локальний процесор та оперативну пам'ять [21].

Інтерфейс користувача та внутрішню логіку маршрутизації побудовано на базі мікрофреймворка Flask [11]. У процесі проектування архітектури розглядалася альтернатива у вигляді Django. Порівняльний аналіз показав, що Django має надлишковий функціонал (вбудовану важку ORM-систему, адміністративну панель), який уповільнює розгортання програми на локальній станції та створює зайве інфраструктурне навантаження. Flask, будучи легковагим інструментом, надає розробнику повний контроль над архітектурою додатка, дозволяє легко ізолювати компоненти та інтегрувати специфічні модулі безпеки, що критично для забезпечення автономності розроблюваної системи.

Компонент збереження відфільтрованих аномальних транзакцій реалізовано за допомогою реляційної системи керування базами даних SQLite. Застосування клієнт-серверних рішень, таких як PostgreSQL або MySQL, було відхилено через їхню архітектурну невідповідність концепції локального автономного інструменту. PostgreSQL вимагає попереднього встановлення та постійного фонового функціонування окремого сервера баз даних, що ускладнює мобільність програмного засобу. SQLite функціонує за принципом вбудованої бібліотеки, зберігаючи всі дані в одному локальному файлі investigations.db, забезпечуючи при цьому повноцінну підтримку транзакцій (ACID) та високу швидкість запису через метод to_sql бібліотеки Pandas.

Безпека збереження накопичених аналітичних даних та захист від несанкціонованого доступу реалізовано засобами пакета werkzeug.security. Для автентифікації користувачів у базі даних не зберігаються паролі у відкритому вигляді, що захищає інформацію у разі компрометації робочої станції дослідника. Замість цього застосовано криптографічне хешування паролів за допомогою алгоритму PBKDF2 з додаванням унікальної солі, що унеможливорює розшифрування або підбір паролів з боку сторонніх осіб. Отриманий стек технологій (Python, Pandas, Flask, SQLite)

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

формує збалансоване, швидкодіє та повністю ізольоване середовище для оперативної обробки блокчейн-даних.

Висновки до першого розділу

У першому розділі було проведено комплексне дослідження теоретичних та правових основ функціонування ринку віртуальних активів в Україні [1], а також проаналізовано існуючі технічні засоби для моніторингу безпеки транзакцій. Розглянуто архітектурні принципи провідних комерційних платформ блокчейн-аналітики, таких як Chainalysis, Elliptic та Nansen, та виявлено їхні ключові обмеження, що полягають у хмарній залежності (SaaS), високій вартості ліцензування та реактивному характері перевірок «за запитом» [13, 21]. На основі аналізу платіжного моніторингу обґрунтовано необхідність створення автономного програмного рішення, яке працює проактивно на локальній станції дослідника та здійснює швидко первинну фільтрацію цілих блоків даних без ризику витоку конфіденційної інформації розслідування.

Визначено числові та топологічні критерії нетиповості транзакцій, які було покладено в основу детермінованих математичних фільтрів для моделей виявлення схем дрібного відмивання коштів («micro») та відстеження переміщення масштабних капіталів інституційних гравців («whale») [12, 14]. На завершення проведено детальний аналіз програмного стеку та обґрунтовано вибір мови Python, бібліотеки Pandas, мікрофреймворка Flask та вбудованої бази даних SQLite. Виявлено, що у порівнянні з альтернативами (Django, PostgreSQL, модулем csv), обраний стек забезпечує високу швидкість векторизованих математичних обчислень над локальними файлами великого обсягу, гнучкість налаштування евристичних фільтрів та повну автономність системи.

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

РОЗДІЛ 2 АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ

2.1 Загальна логіка роботи програми та сценарії використання

Проектування та розробка автономного програмного комплексу для первинного аналізу й фільтрації блокчейн-транзакцій базується на архітектурному шаблоні MVC (Model-View-Controller), адаптованому під специфіку легковагового веб-фреймворка Flask. Програма функціонує повністю в ізольованому локальному середовищі на localhost, що усуває потребу в залученні сторонніх хмарних обчислювальних потужностей і забезпечує конфіденційність транзакційних даних під час проведення досліджень. Вся логіка обробки вхідних файлів зосереджена на безпосередній взаємодії між веб-інтерфейсом, який приймає файли у форматі .tsv, обчислювальними алгоритмами бібліотеки Pandas та локальною базою даних SQLite. Процес обробки є детермінованим і розділяється на кілька послідовних етапів, починаючи від ініціалізації файлу користувачем і закінчуючи фіксацією або видаленням аномальних записів у базі даних investigations.db.

Для розуміння фізичного розміщення компонентів системи та логіки побудови коду доцільно продемонструвати структуру каталогів розробленого проєкту (див. Додаток А). Програмний комплекс має чітку ієрархію папок, де корінь містить головний виконуваний файл app.py, конфігураційні змінні оточення та локальну базу даних. Директорія templates містить набір HTML-шаблонів, які відповідають за рендеринг інтерфейсу головної сторінки, форми автентифікації, профілю користувача, панелі адміністратора, журналу аудиту та дашборду. У папці static розміщено таблиці стилів style.css для забезпечення сучасного візуального відображення та скрипт script.js, який реалізує клієнтське сортування табличних даних без додаткового навантаження на сервер.

Основою взаємодії із системою є два головних сценарії використання, які чітко розмежовують права доступу та функціональні можливості користувачів. Перший сценарій розрахований на загальний або гостьовий доступ і не вимагає проходження процедури автентифікації. Будь-який аналітик може відкрити головну сторінку додатка в браузері, завантажити локальний файл вивантаження транзакцій

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

блокчейну та обрати один із двох вбудованих інструментів аналізу – «micro» або «whale». Після натискання кнопки запуску сканування алгоритм Pandas зчитує файл, проводить обчислення за встановленими математичними фільтрами обсягів, комісій та топології вхідних/вихідних адрес і виводить на екран згенеровану HTML-таблицю із підрахованою кількістю знайдених аномалій. Гостьовий сценарій обмежений лише переглядом і сортуванням результатів, а інтерфейс у цей момент не відображає кнопок для модифікації бази даних, що демонструє скріншот головної сторінки програми в режимі активного сканування.

block_id	hash	time	size	weight	version	lock_time
862479	e7e01e0239f21ade31a8cb1ae998a6d551e1fd01c43e540c9fa3a3fc9ac3b	2024-09-23 06:13:19	341	710	1	0
862496	5ff7a18af9cb8b9f3d7a8ad7d3ee2d3effd492abb841ba52b64482e79220d09e	2024-09-23 08:47:45	382	766	1	0
862500	04127c3fa749d6d8d8c079d6c220e7b76beb6ea81afa8ac3410e2e981ab1236	2024-09-23 09:30:18	382	766	1	0
862500	20eb18a5839c5011022e14fce2451a4765c6b754de783c901ba274567531213a	2024-09-23 09:30:18	382	766	1	0
862502	fb7b012de41dd80d5c198de43da05673236cc8e4335878115069ec5d347a77	2024-09-23 09:45:49	382	766	1	0

Рис.2.1 – Інтерфейс гостьового сценарію

Другий сценарій використання орієнтований на авторизованих фахівців та адміністраторів системи і відкриває повний доступ до управління базою даних та обліковими записами. Після успішного введення логіна та пароля на сторінці автентифікації, сесія користувача фіксується в браузері, і система надає можливість зберегти результати щойно проведеного сканування до таблиці `sus_table` за допомогою кнопки, яка автоматично з'являється над результатами видачі. Авторизований користувач отримує доступ до сторінки «Database Dashboard», де виводиться повний перелік раніше збережених підозрілих транзакцій. Для адміністратора системи біля кожного запису в таблиці дашборду з'являється кнопка видалення, яка викликає кастомне модальне вікно підтвердження дії. Ця функція є критично важливою для

сценарію, коли в процесі детального розслідування аналітик з'ясовує, що транзакція є легітимною, і її необхідно вилучити з реєстру аномалій. Окрім цього, адміністратор має ексклюзивний доступ до панелі модерації для схвалення нових користувачів та до перегляду журналу логів, де фіксується кожна дія в системі, що підтверджується знімками екрана інтерфейсів дашборду та панелі адміністрування.

The screenshot shows the 'Blockchain Analyzer' interface. At the top, there's a header with 'Operator: lexibee (Standard)', 'Database', 'Profile', and 'Logout'. Below this, the 'Upload Block Dataset (.tsv/.csv):' section has a 'Choose File' button and a 'No file chosen' status. The 'Select Threat Model:' dropdown is set to 'Transit / Peeling Chains (Vol. \$1k-\$20k, In/Out ≤ 2, Fee > \$50)'. A large blue 'Scan Network' button is below. The 'Found: 35 anomalies' section features an 'Export to Database' button and a table of anomalies.

block_id	hash	time	size	weight	version	lock
862444	31a39a4ae53e7eda9a49b99d721e0dc8065c3a80e7125baf0a00716743aef1	2024-09-23 00:20:21	632	2528	1	0
862457	80dcbf8cd501d2c5f58127bec447c6ef7afad521ba07153b6f90795358a0fc42	2024-09-23 01:50:31	132822	243006	1	86245
862458	e6ff1a42beb5c37f145c5c6e8682fc8aca2b25d01d0d9922d6d89a0ab64a9a3	2024-09-23 02:00:59	141899	300008	2	0
862467	d0ffe90372c1337888d9b94a2d77427bd262ed0879c2b13b836a52e4568387ed3	2024-09-23 03:45:46	5701	10600	2	7880G
862497	ae08633e8e0f040cecccd17ed3c4d19e5e30724d8de243b87a71eb8120d03ae64	2024-09-23 09:03:58	488	980	1	0

Рис.2.2 – Інтерфейс авторизованого користувача

Saved Anomalies (Total: 44)

	Time	Inputs	Outputs	Input USD	Output USD	Fee USD	Fee %	Model
a029e0d444c9d212ac2a3d96556483261cf7	2024-09-23 19:34:32	1	1	1,019.64	850.05	169.59	0.1663	micro
a4765fc67548e793c901ba274567531213a	2024-09-23 09:30:18	1	2	1,089.32	1,032.11	57.21	0.0525	micro
3effd492abb841ba52b64482e79220d09e	2024-09-23 08:47:45	1	2	1,565.38	1,508.16	57.21	0.0365	micro
4b75b8822e57ae2223081c9ba9359790ae0	2024-09-23 22:38:07	1	1	2,509.48	2,446.55	62.94	0.0251	micro
2ce7f0eca14abafd0c5fcea36bda9c63072	2024-09-23 10:40:33	1	2	2,404.26	2,347.05	57.21	0.0238	micro
65adcb2d84a3fcc5fa071e85b5bb5a0a77e	2024-09-23 19:34:32	1	1	9,711.71	9,590.54	121.16	0.0125	micro
f6a26b1a712416a4374fcbcd80cfe5793a	2024-09-23 19:58:02	1452	1	121,487.03	120,149.19	1,337.84	0.0110	whale
b370f5cf3e59924b9088cb7095b60f78562	2024-09-23 17:46:31	1	1	6,397.06	6,332.26	64.79	0.0101	micro
y5673236c8e4335876115069ec5d347a77	2024-09-23 09:45:49	1	2	6,998.40	6,941.19	57.21	0.0082	micro
aa02c30942f82561d427bbe8856e9a2a451	2024-09-23 17:14:16	178	1	320,017.00	317,855.00	2,162.01	0.0068	whale
a6d551fe1ffd01c43e540c9fa3a3fcfbac3b	2024-09-23 05:13:19	2	1	13,992.81	13,927.70	65.11	0.0047	micro
e7b76beb8ea81afa9ac3410e2e98fab1236	2024-09-23 09:30:18	1	2	13,734.44	13,677.23	57.21	0.0042	micro
34c25f9a048813ad547f810d95b1adb9af0	2024-09-23 19:56:02	243	1	189,845.55	189,472.52	373.03	0.0020	whale

Рис.2.3 – Database Dashboard для користувача без привілеїв адміністратора

Saved Anomalies (Total: 44)

	Time	Inputs	Outputs	Input USD	Output USD	Fee USD	Fee %	Model	Action
c9d212ac2a3d96556483261cf7	2024-09-23 19:34:32	1	1	1,019.64	850.05	169.59	0.1663	micro	Delete
f6e793c901ba274567531213a	2024-09-23 09:30:16	1	2	1,089.32	1,032.11	57.21	0.0525	micro	Delete
841ba52b64482e79220d09e	2024-09-23 08:47:45	1	2	1,565.38	1,508.16	57.21	0.0365	micro	Delete
57ae2223081c9ba9359796ae0	2024-09-23 22:36:07	1	1	2,509.48	2,446.55	62.94	0.0251	micro	Delete
faba9d0c5fcea36bda9c63072	2024-09-23 10:40:33	1	2	2,404.26	2,347.05	57.21	0.0238	micro	Delete
a3fcc5fa071e85b5bb5a0a77e	2024-09-23 19:34:32	1	1	9,711.71	9,590.54	121.16	0.0125	micro	Delete
2416a4374fbcdb80cfe5793a	2024-09-23 19:56:02	1452	1	121,487.03	120,149.19	1,337.84	0.0110	whale	Delete
9924b9088cb7095b60f78562	2024-09-23 17:46:31	1	1	6,397.06	6,332.26	64.79	0.0101	micro	Delete
le4335876115069ec5d347a77	2024-09-23 09:45:49	1	2	6,998.40	6,941.19	57.21	0.0082	micro	Delete
f82561d427bbe8856e9a2a451	2024-09-23 17:14:16	178	1	320,017.00	317,855.00	2,162.01	0.0068	whale	Delete

Рис.2.4 – Database Dashboard для адміністратора

View Audit Logs						
User Management						
ID	Username	Name	Email	Role	Status	Actions
1	admin	System Administrator	admin@system.local	admin	approved	
2	lexibee	Lexy Brown	lexi883@gmail.com	user	approved	Delete
3	dav1dd	David Smith	d4v1dd@gmail.com	user	approved	Delete
4	racercar	Lightning McQueen	racercar@gmail.com	user	approved	Delete
5	addi4ka	Adam Walter	addie40k@ukr.net	user	pending	Approve Delete

Рис.2.5 – Панель модерaції

System Audit Logs			
Log ID	Timestamp	User	Action Details
42	2026-06-10 19:01:57	admin	Approved user ID 5
41	2026-06-10 19:01:23	admin	Logged in successfully
40	2026-06-10 19:01:07	addi4ka	Registered new account (Pending)
39	2026-06-10 18:59:39	admin	Logged out
38	2026-06-10 18:57:47	admin	Logged in successfully
37	2026-06-10 18:57:34	lexibee	Logged out
36	2026-06-10 18:52:58	lexibee	Scanned dataset using whale
35	2026-06-10 18:52:47	lexibee	Scanned dataset using micro
34	2026-06-10 18:52:35	lexibee	Logged in successfully
33	2026-06-10 15:00:17	admin	Logged out
32	2026-06-10 14:59:59	admin	Deleted record: 5b303e1b3a9662aa8788fb397f301767e8e84d62bb89b85934d62483b7b0341f
31	2026-06-10 14:57:43	admin	Approved user ID 4
30	2026-06-10 14:57:34	admin	Logged in successfully

Рис.2.6 – Вміст журналу логів

		БоднаришкА.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

З технічної точки зору, логіка переходів між описаними сценаріями та захист маршрутів реалізовані через систему декораторів фреймворка Flask, які перевіряють наявність ідентифікатора користувача та його ролі в поточній сесії. Головний маршрут обробляє як відображення сторінки, так і приймання файлів через POST-запити. Захищений маршрут збереження результатів аналітики перевіряє авторизацію і, у разі її відсутності, перенаправляє користувача на сторінку логіна. Маршрути перегляду збережених аномалій, видалення записів за конкретними хешами та модерації нових облікових записів чітко розмежовують права, блокуючи несанкціоновані дії користувачів без відповідного статусу, що наочно демонструє базовий каркас архітектурної маршрутизації та логіки контролю доступу в коді програми.

2.2 Розробка алгоритмів фільтрації блокчейн-даних

Алгоритмічна основа розробленого програмного засобу базується на реалізації двох самостійних евристичних моделей аналізу: «micro» та «whale». Кожна модель трансформує теоретичні критерії нетипової поведінки у детерміновані математичні фільтри, що накладаються на завантажений масив даних за допомогою векторизованих методів бібліотеки Pandas. Логіка побудови алгоритмів орієнтована на одночасну перевірку сукупності параметрів транзакції, що дозволяє відсіяти стандартні перекази та ізолювати цільові аномалії без залучення сторонніх хмарних обчислень.

Перший алгоритм реалізує модель «micro», націлену на фіксацію патернів послідовного розщеплення капіталу або автоматизованої бот-активності (так звані схеми «reeling chain»). Математичний апарат цього алгоритму здійснює багаторівневу селекцію транзакцій. На першому етапі відсікаються операції, фіатний еквівалент яких не потрапляє в інтервал від 1 000 до 20 000 доларів США [14]. На другому етапі застосовується топологічний фільтр, який обмежує кількість вхідних і вихідних адрес значенням менше або рівно 2. Це дозволяє виокремити транзакції, де кошти передаються лінійним ланцюжком із мінімальним дробленням. Фінальним критерієм виступає перевірка абсолютної суми комісії, яка має перевищувати 50

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

доларів. Поєднання цих умов дозволяє чітко локалізувати нераціональну з економічного погляду поведінку, притаманну спробам швидкого заплутування слідів обігу активів [13, 21].

Другий алгоритм виконує сканування за моделлю «whale», фокусуючись на виявленні переміщень великих капіталів інституційного рівня, які можуть супроводжувати масштабні процеси виведення кримінальних доходів у фіатні системи [12]. Алгоритм активується за умови, що сумарний обсяг входу транзакції перевищує поріг у 100 000 доларів США. Для відсікання рутинного пакетного трафіку криптовалютних бірж, алгоритм накладає обмеження на кількість вихідних адрес, яка не повинна перевищувати 5. Ця умова відсіює масові автоматизовані розсилки гаманців-кастодіанів. Додатково аналізується розмір комісії: операція маркується як підозріла лише у разі, якщо ініціатор готовий сплатити понад 100 доларів США за її пріоритетне включення у блок.

З погляду програмної реалізації, замість ресурсомістких циклів for мови Python, які суттєво уповільнюють обробку на великих файлах, розроблені алгоритми використовують механізм булевих масок Pandas. Цей підхід дозволяє виконувати фільтрацію масиву на низькому рівні (C-архітектура), порівнюючи цілі стовпці таблиці як вектори за один такт процесора. Після фільтрації алгоритм динамічно збагачує структуру знайдених аномалій, додаючи розрахункову колонку fee_percent (відношення комісії до суми входу) та маркер типу сканування scan_type. Це необхідно для коректної класифікації записів під час їх подальшої фіксації у базі даних.

Нижче наведено програмний уривок, який демонструє безпосередню логіку виконання розроблених алгоритмів фільтрації всередині функції обробки POST-запиту:

```
if scan_type == 'micro':
    sus_transactions = df[(df['input_total_usd'] > 1000) & (df['input_total_usd'] < 20000) & (df['input_count'] <= 2) & (df['output_count'] <= 2) & (df['fee_usd'] > 50)].copy()

elif scan_type == 'whale':
    sus_transactions = df[(df['input_total_usd'] > 100000) & (df['output_count'] <= 5) & (df['fee_usd'] > 100)].copy()
```

		БоднаришкіА.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

Цей уривок додано для демонстрації практичної реалізації логіки відбору, описаної у першому розділі роботи. Він наочно показує, як логічні оператори зв'язують окремі критерії аномальності в єдину маску фільтрації. Код наочно ілюструє повну прозорість алгоритмічного аналізу: на відміну від корпоративних хмарних систем із закритим кодом [13], дослідник може точно простежити, чому конкретний рядок трансляції з файлу .tsv потрапив до фінальної вибірки та отримав статус підозрілого.

На рисунках 2.1, 2.2 продемонстровано фінальний результат роботи алгоритмічного модуля: система вивела точну кількість знайдених аномалій, сформувала інтерактивну таблицю із векторизованим підрахунком відсотків комісії та відкрила для авторизованого оператора функцію експорту цих даних до локальної бази даних.

2.3 Проектування локальної бази даних та захист інформації

Організація збереження виявлених транзакційних аномалій та забезпечення конфіденційності даних розслідувань реалізовані через проектування локальної реляційної бази даних на русії SQLite. Логічна структура сховища орієнтована на забезпечення цілісності інформації, швидку індексацію записів та строге розмежування рівнів доступу суб'єктів системи. Вибір безсерверної архітектури обумовлений необхідністю повної автономності, що унеможливорює перехоплення внутрішнього трафіку бази даних через мережеві сокети, оскільки всі операції читання та запису виконуються безпосередньо у файлі investigations.db у межах операційної системи локальної станції [18].

Архітектура бази даних складається з трьох ключових таблиць: users, sus_table та audit_logs. Таблиця users містить облікові дані операторів, де первинним ключем є унікальний ідентифікатор id. Безпека автентифікації реалізована на рівні програмної логіки фреймворка Flask за допомогою модуля werkzeug.security. Замість збереження паролів у відкритому вигляді, система генерує незворотні криптографічні хеші за алгоритмом PBKDF2 з додаванням індивідуальної солі. Це га-

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

рантує, що навіть за умови повного фізичного копіювання файлу бази даних зломисником, відновлення паролів користувачів суттєво ускладнюється навіть у разі компрометації бази даних. Крім того, таблиця містить атрибут status для реалізації механізму попередньої модерації, що блокує доступ до системи до моменту активації облікового запису адміністратором.

Таблиця sus_table призначена для накопичення результатів евристичного аналізу і використовує унікальний транзакційний hash як первинний ключ. Така схема адресації надійно захищає сховище від дублювання інформації. Перед кожним записом нового масиву даних за допомогою методів Pandas виконується операція порівняння вхідних хешів із уже існуючими в базі. Якщо транзакція вже зафіксована у реєстрі, алгоритм автоматично відсіює її, що запобігає штучному роздуванню обсягу файлу бази даних. Для забезпечення повноцінного ретроспективного аналізу структура таблиці містить дев'ять полів, які повністю повторюють метрики алгоритмів фільтрації, включаючи обсяги, ліміти входів/виходів та маркер моделі загрози scan_type.

Третім компонентом системи захисту та контролю є таблиця audit_logs, яка виконує функцію журналу подій (Audit Log). Будь-яка критична дія в системі – успішний чи невдалий вхід, реєстрація нового профілю, зміна пароля, запуск сканування файлу .tsv чи видалення запису адміністратором – автоматично генерує новий рядок у журналі із фіксацією імені користувача, детального опису дії та точного часового штампа timestamp. Лог-дані записуються виключно в режимі додавання (append-only), а в інтерфейсі програми відсутні будь-які елементи керування для модифікації чи очищення цієї таблиці, що забезпечує фіксацію дій користувачів для подальшого внутрішнього аудиту безпеки.

Нижче наведено програмний уривок SQL-структури, що використовується для ініціалізації таблиць під час першого запуску програми:

```
cursor.execute('''CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY
AUTOINCREMENT, username TEXT UNIQUE NOT NULL, password_hash TEXT NOT NULL, role
TEXT NOT NULL, full_name TEXT, email TEXT, status TEXT NOT NULL)''')
```

Цей уривок демонструє практичну реалізацію реляційної моделі даних та застосування обмежень цілісності на рівні СУБД. Використання модифікатора NOT

		Боднарашик А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

NULL та правила UNIQUE для поля username гарантує неможливість створення дублікатів облікових записів або реєстрації користувачів із порожніми критичними атрибутами, що переносить первинний контроль безпеки на рівень ядра бази даних.

У контексті забезпечення безперервності функціонування системи, архітектура розроблюваного додатка не містить автоматизованих модулів створення резервних копій (бекапів) та процедур гарячого відновлення даних. Таке рішення є обґрунтованим кроком проектування локальних інструментів оперативної аналітики з огляду на два фактори. По-перше, специфіка сховища SQLite полягає в тому, що вся база даних – це один фізичний файл на диску, тому класичне резервне копіювання зводиться до простого дублювання файлу файловою системою ОС, що усуває потребу в роздуванні коду додатка важкими фоновими процесами. По-друге, розроблюваний програмний засіб є інструментом первинного сортування, а не постійним сховищем глобальних даних: першоджерелом аналітики завжди виступають сирі статичні файли .tsv. У разі фізичного пошкодження бази даних, оператор може повторно ініціалізувати файл бази та відновити весь масив аномалій за кілька секунд, просто повторно запустивши алгоритми фільтрації над наявними локальними архівами, що нівелює необхідність утримання складної інфраструктури гарячого резервування.

2.4 Програмна реалізація бекенду та користувацького інтерфейсу

Практична реалізація серверної частини (бекенду) програмного комплексу базується на детермінованій маршрутизації POST та GET-запитів, що зв'язують користувацький інтерфейс із обчислювальними алгоритмами та сховищем даних. Серверний логічний шар розгорнуто в межах одного виконуваного файлу main.py, що спрощує архітектуру додатка та забезпечує високу мобільність локального рішення. Обробка вхідних файлів вивантаження .tsv відбувається в асинхронному режимі без тривалого блокування потоків сервера. Програма приймає об'єкт файлу через глобальний контекст request.files, після чого миттєво передає бінарний потік у конструктор датафрейму Pandas без збереження проміжних копій на жорсткому

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

диску, що мінімізує тимчасовий цифровий слід та підвищує загальну кібербезпеку процесу аналізу.

Користувачський інтерфейс (фронтенд) реалізовано за допомогою шаблонізатора Jinja2, вбудованого у фреймворк Flask. Вебсторінки формуються динамічно на стороні сервера шляхом інтеграції розрахункових масивів у базову розмітку HTML [22, 23]. Візуальна частина адаптована під вимоги лаконічності та ергономіки: всі елементи управління, форми завантаження файлів та випадуючі списки моделей загроз підпорядковані логіці швидкого оперативного використання. Згенерована таблиця результатів не навантажує оперативну пам'ять браузера завдяки оптимізованій конвертації датафрейму через метод `to_html()`. Для забезпечення швидкого аналізу великих списків без повторних запитів до бекенду, на стороні клієнта інтегровано локальний скрипт `script.js`. Цей компонент перехоплює події натискання на заголовки колонок таблиці та миттєво виконує сортування текстових хешів або числових значень обсягів і комісій безпосередньо у вікні браузера.

Для демонстрації реалізації інтерфейсної логіки та зв'язку фронтенду з бекендом нижче наведено програмний уривок коду з файлу головної сторінки `index.html`, який забезпечує динамічний рендеринг результатів сканування:

```
{% if table %}
<div class="results-box">
  <h2>Found: {{ count }} anomalies</h2>
  {% if session.get('user_id') %}
    <form action="/save" method="POST" style="margin-bottom: 15px;">
      <button type="submit" id="save-btn">Export to Database</button>
    </form>
  {% else %}
    <p style="color: #666; font-style: italic;">Log in to save these results
to the database.</p>
  {% endif %}
  <div class="table-container">{{ table | safe }}</div>
</div>
{% endif %}
```

Цей уривок коду ілюструє застосування умовних операторів шаблонізатора для забезпечення контролю рівнів доступу суб'єктів. Блок результатів активується лише за умови наявності згенерованої таблиці, а керуюча кнопка експорту до бази даних `investigations.db` відображається виключно для автентифікованих операторів, які мають активний ідентифікатор сесії `user_id`. Використання фільтра `| safe` вказує

		Боднаришук А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

рушію Jinja2 на те, що рядок таблиці вже є валідованим HTML-кодом, сформованим бібліотекою Pandas, і не потребує додаткового екранування, що забезпечує коректну та швидку візуалізацію даних.

Висновки до другого розділу

У другому розділі роботи детально описано хід проєктування та практичної реалізації автономного програмного комплексу для оперативної аналітики аномальних транзакцій. Сформовано логічну архітектуру додатка на базі локального розгортання мікрофреймворка Flask, що повністю виключає хмарну залежність та мінімізує ризики порушення конфіденційності даних розслідувань. Описано два базові сценарії використання системи, які чітко розмежовують права між анонімними аналітиками (доступ лише до швидкого сканування «тут і зараз») та авторизованими операторами чи адміністраторами системи (функції ведення реєстру транзакцій з ознаками нетипової активності та модерації користувачів).

Розроблено та інтегровано в єдиний програмний комплекс два детерміновані алгоритми фільтрації за моделями «micro» та «whale», які за допомогою векторизованих обчислень бібліотеки Pandas забезпечують швидку обробку великих локальних файлів форматів .tsv за критеріями фіатної вартості, абсолютної комісії та топології адрес. Спроектовано захищену реляційну структуру локальної бази даних SQLite, яка використовує механізми незворотного хешування паролів за алгоритмом PBKDF2 та ведення захищеного від модифікації журналу аудиту (audit_logs). Обґрунтовано відсутність надлишкових інфраструктурних модулів автоматичного бекапування, що оптимізує швидкість роботи додатка. На завершення реалізовано взаємодію бекенду на мові Python із користувацьким інтерфейсом на базі Jinja2 та JavaScript, що дозволило створити швидкодійне, прозоре та повністю ізольоване середовище для первинного фінансового моніторингу блокчейн-даних.

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

РОЗДІЛ 3 ТЕСТУВАННЯ ТА ПРАКТИЧНЕ ВИКОРИСТАННЯ ПРОГРАМНОГО РІШЕННЯ

3.1 Порядок запуску та налаштування локального середовища

Процес розгортання та налаштування локального середовища для функціонування розробленого програмного комплексу орієнтований на забезпечення повної автономності системи без залучення сторонніх хмарних інфраструктур. Першим етапом підготовки є встановлення інтерпретатора мови Python версії 3.11 або вище на робочу станцію дослідника. Після завантаження вихідного коду проєкту у локальну директорію, управління залежностями та ізоляція середовища реалізуються за допомогою вбудованого інструменту `venv` (Virtual Environment). Створення віртуального оточення є критично важливим кроком, який запобігає конфліктам версій бібліотек із глобальними пакетами операційної системи. Активація створеного ізольованого середовища виконується за допомогою стандартних команд термінала операційної системи, що забезпечує перенаправлення всіх подальших викликів пакетного менеджера у внутрішній каталог проєкту.

Наступним кроком є встановлення специфікованих зовнішніх бібліотек. Інсталяція виконується за допомогою утиліти `pip` і включає завантаження та розгортання таких ключових пакетів, як `Flask` для забезпечення маршрутизації та серверної логіки, `pandas` для векторизованої математичної обробки табличних масивів даних та `werkzeug` для криптографічного захисту паролів. Оскільки робота з базою даних `SQLite` не вимагає встановлення окремого сервера чи додаткових зовнішніх драйверів, ядро СУБД ініціалізується автоматично під час першого звернення через стандартну системну бібліотеку `sqlite3` [18]. Конфігурація секретних ключів сесії та налаштування параметрів безпеки середовища виконуються через файл `.env` за допомогою модуля `dotenv`, що дозволяє винести чутливі дані за межі основного коду програми.

Після успішного налаштування залежностей виконується первинний запуск додатка за допомогою виконання команди термінала `python app.py`. Програмний комплекс автоматично розгортає локальний вебсервер `Flask` у режимі відладки

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

(Debug mode: on). Під час першої ініціалізації контексту додатка відпрацьовує вбудована функція `init_db()`. Цей алгоритм перевіряє наявність файлу бази даних `investigations.db` у кореневому каталозі, і у разі його відсутності, самостійно створює таблиці користувачів `users`, виявлених транзакцій `sus_table` та журналу подій `audit_logs`. Автоматично генерується перший обліковий запис адміністратора системи з логіном `admin`, а його пароль миттєво піддається солі та незворотному криптографічному хешуванню, після чого фіксується перший рядок у системному журналі аудиту дій.

```

Run app
C:\PyCharm\python.exe C:\PyCharm\BlockchainTrAnalyser\app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 123-713-655
127.0.0.1 - - [10/Jun/2026 18:43:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Jun/2026 18:43:39] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 18:43:39] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 18:43:51] "POST / HTTP/1.1" 200 -

```

Рис.3.1 – Консоль PyCharm з початковими логами сервера

Особливістю розгортання є те, що після старту сервера сторінка додатка не відкривається в браузері автоматично. Для переходу до графічного інтерфейсу оператор повинен безпосередньо в консолі термінала клікнути на згенероване інтерактивне посилання `http://127.0.0.1:5000`. У цей момент локальний вебсервер Flask приймає перший GET-запит, що миттєво відображається в консолі у вигляді HTTP-логу зі статусом відповіді 200. Шаблонізатор Jinja2 динамічно формує та повертає структуру головної сторінки додатка, паралельно підтягуючи статичні файли стилів `style.css` та клієнтських скриптів `script.js` із кодом відповіді 304, що свідчить про їхнє успішне кешування локальним браузером.

```

127.0.0.1 - - [10/Jun/2026 18:59:41] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:07] "POST /register HTTP/1.1" 302 -
127.0.0.1 - - [10/Jun/2026 19:01:07] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [10/Jun/2026 19:01:07] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:07] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:23] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [10/Jun/2026 19:01:23] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Jun/2026 19:01:23] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:23] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:25] "GET /admin HTTP/1.1" 200 -
127.0.0.1 - - [10/Jun/2026 19:01:25] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:25] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:57] "POST /admin/approve/5 HTTP/1.1" 302 -
127.0.0.1 - - [10/Jun/2026 19:01:57] "GET /admin HTTP/1.1" 200 -
127.0.0.1 - - [10/Jun/2026 19:01:57] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:01:57] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [10/Jun/2026 19:02:01] "GET /admin/logs HTTP/1.1" 200 -

```

Рис.3.2 – Частина звернень до серверної частини

Фінальний етап запуску підтверджує повну готовність локального середовища до проведення оперативного фінансового моніторингу. Будь-яка дія користувача в інтерфейсі – від надсилання форми завантаження .tsv до схвалення нових аналітиків у базі даних через маршрут /admin/approve – обробляється сервером в автономному режимі, фіксуючи часові штампи та типи запитів прямо в консолі. Система забезпечує прозору взаємодію між користувачем та алгоритмами обробки, дозволяючи візуально контролювати стабільність локальних обчислень за логами сервера без ризику виходу конфіденційної інформації за межі робочої станції.

3.2 Тестування алгоритмів класифікації на реальних даних

Для проведення функціонального тестування та оцінки працездатності розроблених алгоритмів фільтрації використано реальний масив транзакційних ончейн-даних, вивантажений у текстовий файл формату .tsv. Кожен з рядків тестового набору даних містить повний технічний профіль окремої транзакції: хеш, обсяг входу у фіатному еквіваленті, суму сплаченої комісії, а також точну кількість вход-

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

них та вихідних адрес. Тестування проводилося локально з метою перевірки математичної коректності відбору аномалій, оцінки швидкодії векторизованих масок Pandas та фіксації результатів обробки безпосередньо у базі даних розслідувань.

block_id	hash	time	size	weight	version	lock_time	is_coinbase	has_witness	input_count	output_count	fee	fee_usd	fee_per_kb	fee_per_kwu
862443	c01c371e1f3a3d80d3449d26c03ad84beb2105c9753c3dc515c83c18a741160d	2024-09-23 00:03:36	269	968	2	0	0	0	0	0	0	0	0	0
862443	3423563a0d9b1194f39773d6883ec5713af621a66d7843575187cf80d09c37e3	2024-09-23 00:03:36	234	936	1	0	0	0	0	0	0	0	0	0
862443	82c30ef80d22c1554eb1e23bcbf04104a4ed35b67ae6138bb10021146f5753f8	2024-09-23 00:03:36	417	1137	2	0	0	0	0	0	0	0	0	0
862443	cf8da2a1151a09334030fb3ee772c5c1af3b96bae5bb8764865fa2a3dd13bae5	2024-09-23 00:03:36	372	837	2	0	0	0	0	0	0	0	0	0
862443	23177c0394c6d2e9b6f65c4ea23f831e5d123784d09aee0178781e99311c15fd	2024-09-23 00:03:36	222	888	2	0	0	0	0	0	0	0	0	0
862443	1b043e4ad31a257b48c9a87a4fcc3c9eab8ad7d6775b7fcfda7d547c66c4e511	2024-09-23 00:03:36	435	975	1	0	0	0	0	0	0	0	0	0
862443	23b2e51d00ac0d386903e13a842c93a2160ff629fbc725eb69c2e2471cfffdd2d	2024-09-23 00:03:36	225	573	2	0	0	0	0	0	0	0	0	0
862443	4e2b7a81e7108268d9aed4f6719e0493af088472a987e1c5410611a1aaf8a62b	2024-09-23 00:03:36	222	888	2	0	0	0	0	0	0	0	0	0

Рис.3.3 – Тестовий набір даних

Перший етап тестування передбачав запуск евристичної моделі «місго» для виявлення ознак послідовного розщеплення капіталу. Після ініціалізації файлу та вибору відповідного типу сканування алгоритм виконав пакетну перевірку всього масиву операцій. Результати обробки показали, що під встановлені критерії (доларовий обсяг від 1 000 до 20 000 доларів, кількість входів та виходів менше або рівно 2, абсолютна комісія понад 50 доларів) потрапили 10 транзакцій. Програма виділила записи, що відповідали встановленим критеріям фільтрації, не включивши до вибірки решту транзакцій досліджуваного блоку. Однак, детальний перегляд виявлених рядків показав майже однакові числові параметри сум надходження та витоку активів. Наприклад, транзакція з обсягом 13 734,44 доларів США, яка мала 1 вхід, 2 виходи та комісію у розмірі 57,21 доларів, за структурою повністю повторювала інші зафіксовані операції. Така повторюваність у межах одного блоку вказує на те, що ці транзакції можуть бути пов'язані з діяльністю фінансових установ

або обмінних сервісів, які автоматично підтримують стабільні ліміти для своїх операцій, а не на реальні спроби розщеплення брудного капіталу кримінальними суб'єктами.

Upload Block Dataset (.tsv/.csv):

Choose File: blockchain_bitcoin_transactions_23092024.tsv

Select Threat Model:

Transit / Peeling Chains (Vol: \$1k-\$20k, In/Out ≤ 2, Fee > \$50)

Transit / Peeling Chains (Vol: \$1k-\$20k, In/Out ≤ 2, Fee > \$50)

High-Volume Consolidation (Vol > \$100k, Outputs ≤ 5)

Рис.3.4 – Опції пошуку нетипових транзакцій

Found: 10 anomalies

Export to Database									
input_count	output_count	input_total	input_total_usd	output_total	output_total_usd	fee	fee_usd	fee_per_kb	fee_per_kb_
2	1	22011303	13,992.81	21908890	13,927.70	102413	65.11	300,331.38	190.92
1	2	21604883	13,734.44	21514883	13,677.23	90000	57.21	235,602.10	149.77
1	1	15276949	9,711.71	15086352	9,590.54	190597	121.16	997,890.06	634.37
1	2	11008801	6,998.40	10918801	6,941.19	90000	57.21	235,602.10	149.77
1	1	10062852	6,397.06	9960932	6,332.26	101920	64.79	530,833.30	337.46
1	1	3947525	2,509.48	3848525	2,446.55	99000	62.94	518,324.60	329.50
1	2	3782007	2,404.26	3692007	2,347.05	90000	57.21	235,602.10	149.77
1	2	2462408	1,565.38	2372408	1,508.16	90000	57.21	235,602.10	149.77
1	2	1713548	1,089.32	1623548	1,032.11	90000	57.21	235,602.10	149.77
1	1	1603932	1,019.64	1337162	850.05	266770	169.59	1,396,701.60	887.90

Рис.3.5 – Виявлені рядки для мікро

Другий етап тестування полягав у перевірці працездатності алгоритму за моделлю «whale» для локалізації переміщень великих капіталів інституційного рівня. Під час сканування масиву даних за детермінованими критеріями макро-аналізу (обсяг понад 100 000 доларів США, кількість отримувачів не більше 5, комісія понад 100 доларів) система виявила 35 операцій, що відповідають заданим технічним параметрам. Проте детальний аналіз вибірки показав, що більшість із них мають майже ідентичні показники вхідних та вихідних сум. Така кореляція обсягів є типовою для внутрішніх операцій великих криптовалютних бірж або кастодіальних сервісів, які здійснюють регулярний ребаланс гаманців, що може свідчити про легітимний характер частини такого трафіку.

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Found: 35 anomalies

Export to Database									
input_count	output_count	input_total	input_total_usd	output_total	output_total_usd	fee	fee_usd	fee_per_kb	fee_per
178	1	503400943	320,017.00	500000000	317,855.00	3400943	2,162.01	128,896.84	81.94
1452	1	191104481	121,487.03	189000000	120,149.19	2104481	1,337.84	9,791.24	6.22
4	1	77243905753	49,104,724.00	77242905753	49,104,088.00	1000000	635.71	1,582,278.50	1,005.8
2	1	77003880380	48,952,136.00	77002880380	48,951,500.00	1000000	635.71	2,958,580.00	1,880.8
2	1	76635507880	48,717,960.00	76634507880	48,717,324.00	1000000	635.71	2,976,190.50	1,891.9
3	1	75923843777	48,265,548.00	75922843777	48,264,912.00	1000000	635.71	2,066,115.90	1,313.4
38	2	1532411099	974,169.06	1531562979	973,629.88	848120	539.16	148,766.89	94.57
243	1	298635464	189,845.55	298048664	189,472.52	586800	373.03	14,107.80	8.97
21	2	1008891087	641,362.12	1008418667	641,061.81	472420	300.32	148,326.53	94.29
28	1	233452709	148,408.22	233028701	148,138.67	424008	269.55	100,690.57	64.01
3	1	320872800	203,982.05	320503800	203,747.47	369000	234.58	756,147.56	480.69
498	1	173236503	110,128.18	172906243	109,918.23	330260	209.95	4,478.34	2.85
20	1	2827821349	1,797,674.25	2827521561	1,797,483.75	299788	190.58	99,465.16	63.23
648	2	500292560	318,040.97	500005769	317,858.66	286791	182.32	2,987.56	1.90
1100	2	655508220	416,713.12	655226504	416,534.03	281716	179.09	1,497.12	0.95

Рис.3.6 – Виявлені рядки для whale

Отримані результати демонструють здатність алгоритму виявляти транзакції, що відповідають заданим критеріям відбору: він безпомилково ідентифікує транзакції з «інституційними» ознаками, проте їх подальша класифікація як «підозрілих» вимагає експертної оцінки адміністратора. Наприклад, транзакція на суму 148 408,22 долара США з одним виходом та комісією 269,55 доларів, хоч і формально підпадає під профіль ризику, за своєю структурою ідентична іншим операціям у вибірці. Це свідчить про те, що алгоритм успішно виконує функцію «первинного сортування», виводячи на екран активність, яка потребує верифікації. У реальних умовах розслідування такий результат не є помилкою, а навпаки – інструментом, що дозволяє аналітику швидко відсіяти масовий корпоративний трафік та видалити його з бази як легітимний, щоб зосередитися на дійсно нетипових одиничних операціях.

Проведена верифікація підтвердила, що програмний засіб коректно обробляє великі масиви даних та надає фахівцю всю необхідну інформацію для прийняття рішення. Роль алгоритму полягає не у фінальному вердикті «злочинна діяльність»,

а у створенні прозорого інструментарію, де адміністратор має змогу критично оцінити результати. Виявлені «схожі» транзакції, що нагадують банківську активність, легко ідентифікуються та видаляються з реєстру аномалій через функціонал інтерфейсу, що забезпечує підтримку чистоти бази даних та фокусування на операціях, які дійсно вимагають глибшого криміналістичного аналізу. Цей процес «очищення» від легітимного корпоративного трафіку є невід’ємною частиною роботи аналітика, а розроблений інструментарій значно пришвидшує цей етап, замінюючи ручну перевірку десятків тисяч рядків автоматизованим відбором за заданими математичними критеріями.

3.3 Аналіз результатів

Аналіз результатів, отриманих у ході функціонального тестування, дозволяє сформулювати об’єктивну оцінку працездатності розробленого програмного комплексу та визначити специфіку структури транзакцій у межах досліджуваного масиву. Критично важливою умовою інтерпретації отриманих даних є той факт, що аналізу піддавався не узагальнений за тривалий час зріз мережевої активності, а один конкретний блок даних, датований 23 вересня 2024 року. Робота з ізольованим блоком фіксованого часового проміжку дає змогу оцінити реальну щільність капіталу та інтенсивність операцій, які відбуваються в екосистемі одномоментно.

Той факт, що обидва алгоритми разом виділили лише 45 транзакцій, свідчить про доцільність обраних початкових параметрів математичних фільтрів та їхню здатність відсіювати основний масив щоденного користувацького шуму у межах досліджуваного масиву, спрощуючи аналітику подальший процес перевірки даних.

Проте найважливішим результатом аналізу є якісна однорідність виявлених аномалій, яка стала очевидною під час перегляду збереженого реєстру. Як у моделі макро-переказів, так і в моделі дрібного розщеплення капіталу, переважна більшість прапорців (flagged transactions) вказувала на операції з майже однаковими сумами надходження та витоку активів. У блоці від 23 вересня 2024 року ця тенденція простежується системно. Така математична повторюваність параметрів і балансів

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

може розглядатися як одна з ознак автоматизованої діяльності регульованих фінансових установ, криптовалютних бірж чи сервісів інституційного кастодіального зберігання.

Для детального розуміння логіки сортування та фіксації таких повторюваних результатів адміністратором системи нижче наведено знімок екрана захищеного інтерфейсу бази даних, де накопичуються результати аналізу.

Blockchain Analyzer										Operator: admin (Privileged)		Database	Admin Panel	Profile	Logout
Saved Anomalies (Total: 38)															
	Time	Inputs	Outputs	Input USD	Output USD	Fee USD	Fee %	Model	Action						
lc9d212ac2a3d98556483261cf7	2024-09-23 19:34:32	1	1	1,019.64	850.05	169.59	0.1663	micro	Delete						
48e793c901ba274567531213a	2024-09-23 09:30:18	1	2	1,089.32	1,032.11	57.21	0.0525	micro	Delete						
3841ba52b64482e79220d09e	2024-09-23 08:47:45	1	2	1,565.38	1,508.16	57.21	0.0365	micro	Delete						
357ae2223081c9ba9359796ae0	2024-09-23 22:38:07	1	1	2,509.48	2,446.55	62.94	0.0251	micro	Delete						
4abaf90c5fcea38bda9c63072	2024-09-23 10:40:33	1	2	2,404.26	2,347.05	57.21	0.0238	micro	Delete						
1a3fcc5fa071e85b5bb5a0a77e	2024-09-23 19:34:32	1	1	9,711.71	9,590.54	121.16	0.0125	micro	Delete						
2416a4374fcbcd80cda5793a	2024-09-23 19:58:02	1452	1	121,487.03	120,149.19	1,337.84	0.0110	whale	Delete						
39924b9068cb7095b60f78562	2024-09-23 17:48:31	1	1	6,397.06	6,332.26	64.79	0.0101	micro	Delete						
8e4335876115069ec5d347a77	2024-09-23 09:45:49	1	2	6,998.40	6,941.19	57.21	0.0082	micro	Delete						
f62561d427bbe8850e9a2a451	2024-09-23 17:14:16	178	1	320,017.00	317,855.00	2,162.01	0.0068	whale	Delete						

Рис.3.7 – Інтерфейс перегляду та адміністрування збережених транзакційних аномалій

Оцінюючи результати моделі «whale», де було зафіксовано транзакції обсягом понад 100 000 доларів США із обмеженою кількістю отримувачів, аналітик може розглядати такі результати як можливий приклад корпоративного ребалансу ліквідності. Біржі дбають про власну репутацію та суворо дотримуються внутрішніх технічних регламентів безпеки, тому їхні транзакції мають чіткі, циклічні й шаблонні фінансові параметри. Збіг сум у вивантаженні з точністю до centa вказує на переміщення коштів між холодними та гарячими гаманцями платформи, що не дозволяє однозначно пов'язати такі операції з протиправною діяльністю, незважаючи на їхній величезний обсяг.

Аналогічна логіка спостерігається і в результатах роботи алгоритму «micro». Шаблонні транзакції в діапазоні від 1 000 до 20 000 доларів США із двома виходами

		Боднаришук А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

та фіксованою завищеною комісією, які за технічними ознаками нагадують схеми розщеплення («peeling chain»), за структурою розподілу балансів ймовірно відображають автоматизовану роботу легітимних обмінних пунктів.

З погляду практичного розслідування кіберзлочинів, такий результат сканування окремого блоку є закономірним та інформативним. Блокчейн є відкритим середовищем, де комерційний трафік складає левову частку активності. Роль розробленого програмного засобу полягає в тому, щоб миттєво підняти цей масив на поверхню та структурувати його. Виявлення ідентичних патернів дозволяє адміністратору зробити швидкий аналітичний висновок про легітимність походження цих грошових потоків і одним кліком видалити безпечну транзакцію через функціонал дашборду. Це підтверджує, що розроблена система не приймає остаточного рішення щодо легітимності чи протиправності транзакцій. Вона виконує функцію первинного експрес-відбору потенційно підозрілих операцій із великого масиву сирих даних. Отримані результати призначені для використання аналітиками інформаційної безпеки, фахівцями центрів моніторингу (SOC-аналітиками) та спеціалістами з фінансових розслідувань як надійне підґрунтя для подальшого детального аналізу.

Усунення інформаційного шуму, створеного біржами та обмінниками, значно звужує коло подальшого пошуку. Після того як шаблонна активність фінансових інституцій маркується як безпечна, в аналітичній вибірці залишаються лише поодинокі, дійсно атипові транзакції, які не мають системних повторень. Саме ці унікальні відхилення, що випадають із загальної логіки автоматизованих скриптів комерційних платформ, і стають головними цілями для глибокого криміналістичного аналізу та деанонізації [13].

Додатковим фактором успішного тестування є підтвердження високої надійності системи захисту інформації, інтегрованої в архітектуру додатка. Під час інтенсивної роботи з базою даних, видалення та збереження записів, кожен крок оператора безпомилково фіксувався у внутрішньому журналі подій. Спроби неавторизованого доступу до сторінки адміністратора чи дашборду бази даних блокувалися

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

системою на рівні маршрутизації бекенду, а записи в таблиці audit_logs чітко відображали стабільність контролю доступу суб'єктів. Для візуального підтвердження безперервного моніторингу дій у системі та перевірки цілісності цифрового сліду під час тестування нижче наведено знімок екрана системного журналу.

System Audit Logs			
Log ID	Timestamp	User	Action Details
68	2026-06-10 20:36:27	admin	Logged in successfully
67	2026-06-10 20:36:02	admin	Logged out
66	2026-06-10 20:17:19	admin	Saved 10 anomaly records
65	2026-06-10 20:17:18	admin	Scanned dataset using micro
64	2026-06-10 20:17:00	admin	Deleted record: 0260b28c799a2cbd35a9a18f2da76c6cd3320e8ea722eda01b4bdea77e994175
63	2026-06-10 20:16:56	admin	Deleted record: bf443847e33140988b1543440d80e029e0d444c9d212ac2a3d96556483261cf7
62	2026-06-10 20:16:31	admin	Deleted record: ae08633e8e0f040cecc0d17ed3c4d19e5e30724d8de243b67a71eb6120d03ae64
61	2026-06-10 20:16:15	admin	Deleted record: dc8e90372c1337888d9b94a2d77427bd262ed0879c2813b936a52e4568387ed3
60	2026-06-10 20:16:09	admin	Deleted record: e6ff1a42beb5c37f1145c5c6e6682fc9aca2b25d01d0d9922d6df89a0ab64a9a3
59	2026-06-10 20:16:06	admin	Deleted record: 8cdcbf8cd501d2c5f58127bac447c0ef7afad521ba07f53b6f90795358a0fc42
58	2026-06-10 20:16:02	admin	Deleted record: 15bab4f02dfe4890bb0c0ac00b6b4b75b8822e57ae2223081c9ba9359796ae0
57	2026-06-10 20:14:35	admin	Deleted record: 47b1d17a5bc935313b2f06fba586265adcb2d84a3fcc5fa071e85b5bb5a0a77e
56	2026-06-10 20:14:27	admin	Deleted record: ec598fd54832d4030988107df1a7db370f5cf3e59924b9088cb7095b60f78562

Рис.3.8 – Журнал аудиту безпеки та фіксації дій користувачів у системі

На рисунку 3.8 продемонстровано структуру журналу подій системи, який є захисним каркасом програмного засобу та підтверджує його відповідність вимогам безпеки при роботі з чутливими даними розслідувань.

Підводячи підсумок аналітичної оцінки, можна стверджувати, що розроблений програмний засіб повністю виконав покладені на нього завдання первинного сортування. Незважаючи на те, що всі виявлені у блоці від 23 вересня 2024 року аномалії мали ознаки легітимного корпоративного трафіку фінансових установ, тестування довело головне: алгоритми здатні миттєво локалізувати задані патерни в масі сирих даних. Прозорість налаштування фільтрів та локальний характер обробки забезпечують аналітику повний контроль над ходом дослідження, надаючи надійний і швидкий інструмент першої лінії захисту від фінансових зловживань у криптосфері.

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

Висновки до третього розділу

У третьому розділі роботи проведено успішне розгортання, практичне налаштування та функціональне тестування розробленого програмного комплексу в ізолюваному локальному середовищі на базі віртуального оточення Python. Описано покроковий порядок ініціалізації додатка, запуску локального вебсервера Flask та взаємодії оператора з інтерактивним посиланням у консолі термінала. На основі аналізу логів сервера підтверджено стабільність обробки HTTP-запитів, швидкість завантаження статичних файлів стилів та скриптів сортування, а також автоматичне створення реляційних таблиць бази даних SQLite при першому старті системи.

Проведено функціональне тестування алгоритмів класифікації на реальному ончейн-масиві даних, що містив один фіксований блок від 23 вересня 2024 року. Аналіз результатів виявив якісну однорідність та шаблонність більшості зафіксованих операцій, що дає підстави припустити їхню належність до легітимної діяльності криптовалютних бірж та великих обмінних сервісів, які підтримують стабільні ліміти для захисту власної репутації.

Доведено, що розроблене рішення ефективно виконує функцію швидкого фільтра першого рівня, дозволяючи аналітику моментально ідентифікувати автоматизований корпоративний трафік, оперативно очистити від нього базу даних через захищений інтерфейс дашборду та зосередитися на поодиноких унікальних аномаліях, які дійсно потребують розслідування. Верифікація роботи системи захисту інформації засвідчила надійність криптографічного хешування облікових даних та працездатність механізму журналювання подій безпеки, що вказує на перспективність подальшої практичної апробації програмного комплексу в задачах кібербезпеки та фінансового моніторингу віртуальних активів.

ВИСНОВКИ

У кваліфікаційній роботі розроблено та протестовано автономний програмний комплекс для проактивного фінансового моніторингу та первинної фільтрації транзакційних аномалій у блокчейн-мережах. У ході дослідження повністю досягнуто поставленої мети та реалізовано всі визначені завдання. На основі аналізу нормативно-правової бази та теоретичних засад функціонування ринку віртуальних активів в Україні було чітко структуровано класифікацію забезпечених і незабезпечених цифрових інструментів. Проведений порівняльний аналіз архітектури провідних світових комерційних платформ, таких як Chainalysis, Elliptic та Nansen, дозволив виявити їхні суттєві обмеження, пов'язані із високою вартістю ліцензування, залежністю від хмарних сервісів (SaaS) та реактивним характером перевірок адреси за запитом. Це дало змогу обґрунтувати актуальність створення легковагового локального інструменту, що працює проактивно безпосереднього на робочій станції аналітика, гарантуючи повну конфіденційність ходу розслідувань.

Для виявлення загроз формалізовано критерії нетиповості транзакцій та розроблено дві евристичні моделі: «micro» (дрібне розщеплення капіталу) та «whale» (масштабні переміщення). Фінансові пороги для них (1 000-20 000 та понад 100 000 доларів США відповідно) визначено експериментальним шляхом для мінімізації хибних спрацювань. За результатами порівняння альтернатив (Django, PostgreSQL) обрано програмний стек: Python, Pandas, Flask та SQLite. Цей комплекс забезпечує високу швидкість обробки локальних файлів та має мінімальні інфраструктурні вимоги.

У межах виконання практичного етапу роботи спроектовано та реалізовано внутрішню логіку автономного програмного засобу, що працює в ізольованому середовищі localhost. Розроблено два сценарії використання системи, які чітко розмежовують права доступу між анонімним дослідником (оперативне сканування блоків) та авторизованим оператором, для якого реалізовано функції ведення захищеного реєстру виявлених нетипових операцій. Спроектовано реляційну структуру локальної бази даних investigations.db, захист інформації в якій реалізовано на рівні ядра через механізми незворотного криптографічного хешування облікових даних

		Боднариш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

користувачів за алгоритмом PBKDF2 та ведення журналу аудиту безпеки (audit_logs). Обґрунтовано відсутність надлишкових інфраструктурних модулів автоматичного бекапування, оскільки специфіка збереження даних у вигляді єдиного фізичного файлу та наявність сирих файлів-першоджерел дозволяють відновити реєстр аномалій шляхом повторного запуску фільтрації над вихідними даними.

Практичне тестування розроблених алгоритмів класифікації проводилося на реальному ончейн-масиві даних, що містив один фіксований блок від 23 вересня 2024 року. Якісний аналіз сформованої вибірки виявив високу повторюваність та шаблонність числових показників більшості зафіксованих операцій, що вказує на їхній ймовірний зв'язок із легітимною автоматизованою діяльністю криптовалютних бірж та великих обмінних сервісів, які здійснюють регулярний ребаланс гаманців.

Результати тестування показали, що розроблене рішення може використовуватися як інструмент первинного відбору транзакцій з ознаками нетипової активності для подальшого аналізу, надаючи аналітику можливість налаштування критеріїв відбору. Спроектований комплекс орієнтований виключно на попереднє сортування даних і не виносить фінального вердикту про легітимність транзакцій. Результати тестування засвідчили стабільність інтерфейсу та надійність системи контролю доступу суб'єктів. Програмне рішення демонструє готовність до подальшого дослідження та практичної апробації підрозділами кібербезпеки, а результати його роботи можуть аналізуватися фахівцями з фінансових розслідувань та SOC-аналітиками для первинного оцінювання великих масивів даних у сфері обігу віртуальних активів.

		Боднарашик А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Про віртуальні активи : Закон України від 17.02.2022 № 2074-IX : станом на 15 листоп. 2024 р. URL: <https://zakon.rada.gov.ua/laws/show/2074-20#Text> (дата звернення: 04.06.2026).
2. Про електронну комерцію : Закон України від 03.09.2015 № 675-VIII : станом на 1 січ. 2024 р. URL: <https://zakon.rada.gov.ua/laws/show/675-19#Text> (дата звернення: 04.06.2026).
3. Про платіжні послуги : Закон України від 30.06.2021 № 1591-IX : станом на 26 груд. 2025 р. URL: <https://zakon.rada.gov.ua/laws/show/1591-20#Text> (дата звернення: 04.06.2026).
4. Криптовалюта у кримінальних провадженнях і не тільки. *Юридична газета – онлайн версія*. URL: <https://yur-gazeta.com/publications/practice/kriminalne-pravo-ta-proces/kriptovalyuta-u-kriminalnih-provadhennyah-i-ne-tilki.html> (дата звернення: 04.06.2026).
5. Криптовалютні злочини щороку відбирають в Україні мільярди втрачених доходів – звіт RUSI. *Mind.ua*. URL: <https://mind.ua/news/20293934-kriptovalyutni-zlochiny-shchoroku-vidbirayut-v-ukrayini-milyardi-vtrachenih-dohodiv-zvit-rusi> (дата звернення: 04.06.2026).
6. Обман у крипті: у кіберполіції розповіли про ключові загрози та нові схеми шахрайства. *Департамент Кіберполіції*. URL: <https://cyberpolice.gov.ua/news/obman-u-krypti-u-kiberpolicziyi-rozpovily-pro-klyuchovi-zagrozy-ta-novi-sxemy-shaxrajstva-1763/> (дата звернення: 04.06.2026).
7. Радева А. Криптовалюта під прицілом судів: від арешту до поділу майна. *Liga Zakon*. URL: https://biz.ligazakon.net/analitics/240354_kriptovalyutapd-priyslom-sudv-vd-areshtu-do-podlu-mayna (дата звернення: 04.06.2026).
8. Andy Greenberg on how 'Tracers in the Dark' found the dark web's worst criminals. *CyberScoop*. URL: <https://cyberscoop.com/irs-criminal-investigations-cryptocurrency/> (дата звернення: 04.06.2026).

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

9. Blockchain

Structure. *GeeksforGeeks*.

URL: <https://www.geeksforgeeks.org/ethical-hacking/blockchain-structure/> (дата звернення: 04.06.2026).

10. FBI nominee Kash Patel gets questions on cybercrime investigations, Silk Road founder, surveillance powers. *CyberScoop*. URL: <https://cyberscoop.com/fbi-nominee-kash-patel-cybercrime-silk-road-surveillance-powers/> (дата звернення: 04.06.2026).

11. Flask

Documentation

(3.1.x). *Flask*.

URL: <https://flask.palletsprojects.com/en/stable/> (дата звернення: 04.06.2026).

12. How Illicit Actors Launder Money Through Crypto Exchanges. *sanctions.io*. URL: <https://www.sanctions.io/blog/how-illicit-actors-launder-money-through-crypto-exchanges> (дата звернення: 04.06.2026).

13. How to squeeze investigative evidence from any cryptocurrency address. *Elliptic*. URL: <https://www.elliptic.co/blog/how-to-squeeze-investigative-evidence-from-any-cryptocurrency-address> (дата звернення: 04.06.2026).

14. Money laundering through cryptocurrencies. *United Nations*. URL: <https://syntheticdrugs.unodc.org/syntheticdrugs/en/cybercrime/laundryingproceeds/moneylaundering.html> (дата звернення: 04.06.2026).

15. Previously unknown crypto address of Silk Road founder Ross Ulbricht found. *Medium.com*. URL: <https://medium.com/@AMLCrypto/previously-unknown-crypto-address-of-silk-road-founder-ross-ulbricht-found-972e0cb47d3f> (дата звернення: 04.06.2026).

16. Silk Road Founder's \$47 Million BTC Wallet Becomes Active After 10 Years. *Binance*. URL: <https://www.binance.com/en/square/post/19297619968241> (дата звернення: 04.06.2026).

17. Silk Road. *Britannica.com*. URL: <https://www.britannica.com/topic/Silk-Road-marketplace> (дата звернення: 04.06.2026).

18. SQL As Understood By SQLite. *sqlite.org*. URL: <https://sqlite.org/lang.html> (дата звернення: 04.06.2026).

		Боднарашик А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

19. Susnjara S., Smalley I. What Is Blockchain? *IBM*.
URL: <https://www.ibm.com/think/topics/blockchain> (дата звернення: 04.06.2026).

20. The dark-web market Silk Road. *Avira Blog*.
URL: <https://www.avira.com/en/blog/dark-web-black-market-silk-road?srsId=AfmBOorB2TkGZKPH75wii0nLdN-hSngcuuiDGuxsuJNAu4ubYxCPrFt> (дата звернення: 04.06.2026).

21. What Is Transaction Clustering in Crypto? *Nansen AI*.
URL: <https://nansen.ai/post/what-is-transaction-clustering-in-crypto-address-analysis> (дата звернення: 04.06.2026).

22. Creating a Login Page with Python Flask, HTML, CSS, and JavaScript. *Medium*. URL: <https://almufid.medium.com/creating-a-login-page-with-python-flask-html-css-and-javascript-afd1f890bb87> (дата звернення: 04.06.2026).

23. How to build a web application using Flask and deploy it to the cloud. *FreeCodeCamp*. URL: <https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/> (дата звернення: 04.06.2026).

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

ДОДАТКИ

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

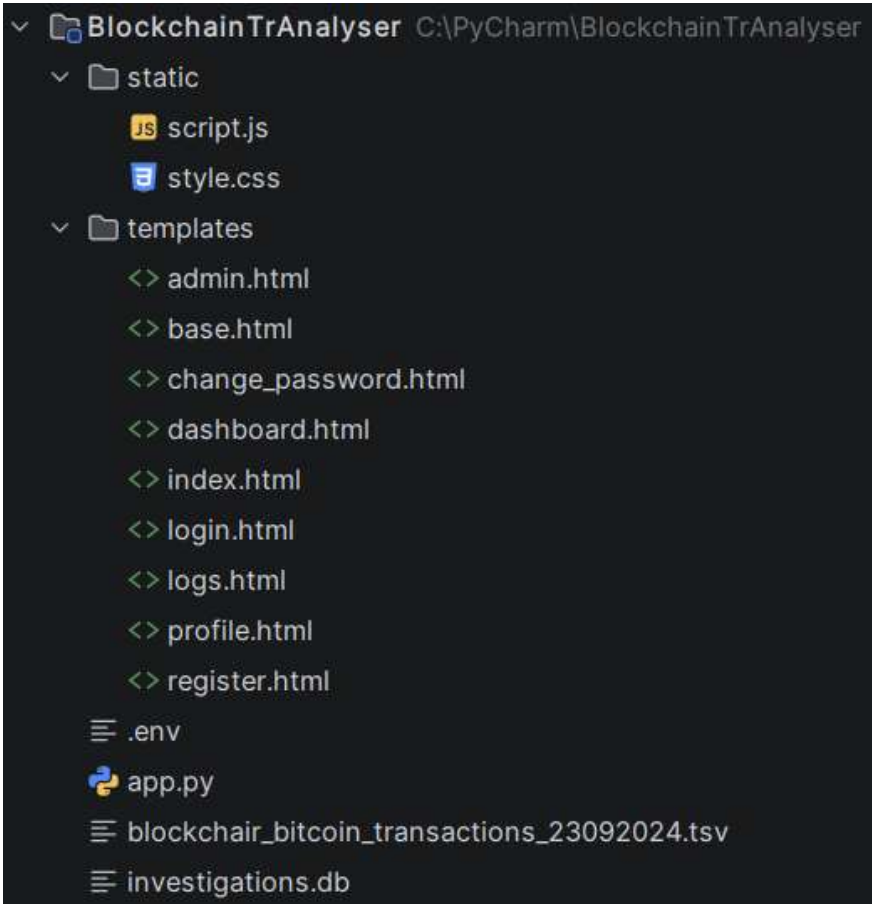


Рис.1.А. Структура каталогу програмного комплексу та розташування компонентів проєкту

Лістинг коду програмного рішення

Python:

```

from flask import Flask, render_template, request, redirect, url_for, flash, session
import pandas as pd
import sqlite3
import os
import re
from datetime import datetime
from dotenv import load_dotenv
from werkzeug.security import generate_password_hash, check_password_hash

load_dotenv()

app = Flask(__name__)
app.secret_key = os.getenv("MY_SECRET_KEY", "FallbackKey123")
temp_data = None

def check_password_strength(password):
    if len(password) < 8: return False, "Password must be at least 8 characters long."
    if not re.search(r"[a-z]", password): return False, "Password must contain at least one lowercase letter."
    if not re.search(r"[A-Z]", password): return False, "Password must contain at least one uppercase letter."
    if not re.search(r"[0-9]", password): return False, "Password must contain at least one number."
    return True, ""

def log_action(username, action):
    conn = sqlite3.connect('investigations.db')
    cursor = conn.cursor()
    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    cursor.execute("INSERT INTO audit_logs (username, action, timestamp) VALUES (?, ?, ?)", (username, action, now))
    conn.commit()
    conn.close()

def init_db():
    conn = sqlite3.connect('investigations.db')
    cursor = conn.cursor()

    cursor.execute('''CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT UNIQUE NOT NULL,
        password_hash TEXT NOT NULL,
        role TEXT NOT NULL, full_name TEXT, email TEXT, status TEXT NOT NULL)''')

    cursor.execute('''CREATE TABLE IF NOT EXISTS sus_table (
        hash TEXT PRIMARY KEY, time TEXT, input_count INTEGER, output_count INTEGER,
        input_total_usd REAL, output_total_usd REAL, fee_usd REAL, fee_percent REAL, scan_type TEXT)''')

    cursor.execute('''CREATE TABLE IF NOT EXISTS audit_logs (
        id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT, action TEXT, timestamp TEXT)''')

```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

        cursor.execute("SELECT COUNT(*) FROM users")
        if cursor.fetchone()[0] == 0:
            hashed_pw = generate_password_hash("Admin123!")
            cursor.execute(
                "INSERT INTO users (username, password_hash, role, full_name, email,
status) VALUES (?, ?, ?, ?, ?, ?)",
                ("admin", hashed_pw, "admin", "System Administrator",
"admin@system.local", "approved"))

            now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            cursor.execute("INSERT INTO audit_logs (username, action, timestamp)
VALUES (?, ?, ?)",
                           ("SYSTEM", "Created default admin account", now))

        conn.commit()
        conn.close()

with app.app_context():
    init_db()

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        conn = sqlite3.connect('investigations.db')
        cursor = conn.cursor()
        cursor.execute("SELECT id, username, password_hash, role, status FROM
users WHERE username = ?", (username,))
        user = cursor.fetchone()
        conn.close()

        if user and check_password_hash(user[2], password):
            if user[4] != 'approved':
                log_action(username, "Failed login (Pending approval)")
                flash("Your account is pending admin approval.")
                return redirect(url_for('login'))

            session['user_id'] = user[0]
            session['username'] = user[1]
            session['role'] = user[3]
            log_action(username, "Logged in successfully")
            flash("Authentication successful.")
            return redirect(url_for('home'))
        else:
            log_action(username, "Failed login (Invalid credentials)")
            flash("Access Denied: Invalid credentials.")
        return render_template('login.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        full_name = request.form['full_name']
        email = request.form['email']

        is_valid, msg = check_password_strength(password)
        if not is_valid:

```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

        flash(f"Security Requirement: {msg}")
        return redirect(url_for('register'))

conn = sqlite3.connect('investigations.db')
cursor = conn.cursor()
try:
    hashed_pw = generate_password_hash(password)
    cursor.execute(
        "INSERT INTO users (username, password_hash, role, full_name,
email, status) VALUES (?, ?, ?, ?, ?, ?)",
        (username, hashed_pw, "user", full_name, email, "pending"))

    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    cursor.execute("INSERT INTO audit_logs (username, action, timestamp)
VALUES (?, ?, ?)",
                    (username, "Registered new account (Pending)", now))

    conn.commit()
    flash("Registration successful! Wait for Admin approval.")
    return redirect(url_for('login'))
except sqlite3.IntegrityError:
    flash("Username already taken.")
finally:
    conn.close()

return render_template('register.html')

@app.route('/logout')
def logout():
    if 'username' in session: log_action(session['username'], "Logged out")
    session.clear()
    flash("You have been securely logged out.")
    return redirect(url_for('home'))

@app.route('/', methods=['GET', 'POST'])
def home():
    global temp_data
    if request.method == 'POST':
        uploaded_file = request.files['dataset']
        scan_type = request.form.get('scan_type')

        if uploaded_file:
            df = pd.read_csv(uploaded_file, sep='\t')
            df['fee_percent'] = df['fee_usd'] / df['input_total_usd']

            if scan_type == 'micro':
                sus_transactions = df[(df['input_total_usd'] > 1000) &
(df['input_total_usd'] < 20000) &
                                     (df['input_count'] <= 2) &
(df['output_count'] <= 2) & (
                                     df['fee_usd'] > 50)].copy()
            elif scan_type == 'whale':
                sus_transactions = df[
(df['input_total_usd'] > 100000) & (df['output_count'] <= 5) &
(df['fee_usd'] > 100)].copy()
            else:
                sus_transactions = pd.DataFrame()

            if not sus_transactions.empty:
                sus_transactions['scan_type'] = scan_type

        temp_data = sus_transactions

```

		БоднарашикА.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

        html_table = sus_transactions.to_html(index=False, float_format=lambda
x: f"{x:,.2f}", classes='data-table')
        if 'username' in session: log_action(session['username'], f"Scanned
dataset using {scan_type}")
        return render_template('index.html', table=html_table,
count=len(sus_transactions))
        return render_template('index.html', table=None, count=0)

@app.route('/save', methods=['POST'])
def save_to_db():
    if 'user_id' not in session: return redirect(url_for('login'))
    global temp_data

    if temp_data is not None and not temp_data.empty:
        conn = sqlite3.connect('investigations.db')
        columns = ['hash', 'time', 'input_count', 'output_count',
'input_total_usd', 'output_total_usd', 'fee_usd',
'fee_percent', 'scan_type']
        data_to_save = temp_data[columns]
        try:
            old_hashes = pd.read_sql("SELECT hash FROM sus_table",
conn)['hash'].tolist()
            new_data = data_to_save[~data_to_save['hash'].isin(old_hashes)]
        except Exception:
            new_data = data_to_save

        if not new_data.empty:
            new_data.to_sql('sus_table', conn, if_exists='append', index=False)

            now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            conn.execute("INSERT INTO audit_logs (username, action, timestamp)
VALUES (?, ?, ?)",
                        (session['username'], f"Saved {len(new_data)} anomaly
records", now))
            conn.commit()

            flash(f"Secured {len(new_data)} new records.")
        else:
            flash("All anomalies are already logged.")
        conn.close()
    else:
        flash("No data available to save.")
    return redirect(url_for('home'))

@app.route('/dashboard')
def dashboard():
    if 'user_id' not in session: return redirect(url_for('login'))
    conn = sqlite3.connect('investigations.db')
    df = pd.read_sql("SELECT * FROM sus_table", conn)
    records = df.to_dict(orient='records')
    conn.close()
    return render_template('dashboard.html', records=records)

@app.route('/delete/<tx_hash>', methods=['POST'])
def delete_record(tx_hash):
    if session.get('role') != 'admin': return redirect(url_for('dashboard'))

    conn = sqlite3.connect('investigations.db')
    conn.execute("DELETE FROM sus_table WHERE hash = ?", (tx_hash,))

    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

```

        conn.execute("INSERT INTO audit_logs (username, action, timestamp) VALUES (?,
?, ?)",
                    (session['username'], f"Deleted record: {tx_hash}", now))

        conn.commit()
        conn.close()

        flash("Record deleted successfully.")
        return redirect(url_for('dashboard'))

@app.route('/profile', methods=['GET', 'POST'])
def profile():
    if 'user_id' not in session: return redirect(url_for('login'))
    conn = sqlite3.connect('investigations.db')
    cursor = conn.cursor()

    if request.method == 'POST':
        new_username, new_full_name, new_email = request.form['username'],
        request.form['full_name'], request.form[
            'email']
        try:
            cursor.execute("UPDATE users SET username=?, full_name=?, email=?
WHERE id=?",
                            (new_username, new_full_name, new_email,
session['user_id']))
            conn.commit()
            if new_username != session['username']:
                now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
                cursor.execute("INSERT INTO audit_logs (username, action,
timestamp) VALUES (?, ?, ?)",
                                (session['username'], f"Changed username to
{new_username}", now))
            conn.commit()
            session['username'] = new_username
            flash("Profile updated successfully.")
        except sqlite3.IntegrityError:
            flash("Username already taken.")

        cursor.execute("SELECT username, full_name, email FROM users WHERE id=?",
            (session['user_id'],))
        user_info = cursor.fetchone()
        conn.close()
        return render_template('profile.html', user=user_info)

@app.route('/change_password', methods=['GET', 'POST'])
def change_password():
    if 'user_id' not in session: return redirect(url_for('login'))
    if request.method == 'POST':
        current_password, new_password, confirm_password =
        request.form['current_password'], request.form[
            'new_password'], request.form['confirm_password']

        conn = sqlite3.connect('investigations.db')
        cursor = conn.cursor()
        cursor.execute("SELECT password_hash FROM users WHERE id = ?",
            (session['user_id'],))
        user = cursor.fetchone()

        if not check_password_hash(user[0], current_password):
            conn.close()
            flash("Incorrect current password.")
            return redirect(url_for('change_password'))

```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

```

        if new_password != confirm_password:
            conn.close()
            flash("New passwords do not match.")
            return redirect(url_for('change_password'))

        is_valid, msg = check_password_strength(new_password)
        if not is_valid:
            conn.close()
            flash(f"Security Requirement: {msg}")
            return redirect(url_for('change_password'))

        hashed_pw = generate_password_hash(new_password)
        cursor.execute("UPDATE users SET password_hash = ? WHERE id = ?",
(hashed_pw, session['user_id']))

        now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        cursor.execute("INSERT INTO audit_logs (username, action, timestamp)
VALUES (?, ?, ?)",
                      (session['username'], "Changed password", now))

        conn.commit()
        conn.close()
        flash("Password updated successfully.")
        return redirect(url_for('profile'))

    return render_template('change_password.html')

@app.route('/admin', methods=['GET'])
def admin_panel():
    if session.get('role') != 'admin': return redirect(url_for('home'))
    conn = sqlite3.connect('investigations.db')
    cursor = conn.cursor()
    cursor.execute("SELECT id, username, full_name, email, role, status FROM
users")
    all_users = cursor.fetchall()
    conn.close()
    return render_template('admin.html', users=all_users)

@app.route('/admin/approve/<int:user_id>', methods=['POST'])
def approve_user(user_id):
    if session.get('role') == 'admin':
        conn = sqlite3.connect('investigations.db')
        conn.execute("UPDATE users SET status='approved' WHERE id=?", (user_id,))

        now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        conn.execute("INSERT INTO audit_logs (username, action, timestamp) VALUES
(?, ?, ?)",
                      (session['username'], f"Approved user ID {user_id}", now))

        conn.commit()
        conn.close()
        flash("User approved.")
        return redirect(url_for('admin_panel'))

@app.route('/admin/delete_user/<int:user_id>', methods=['POST'])
def delete_user(user_id):
    if session.get('role') == 'admin':
        if user_id == session['user_id']:
            flash("You cannot delete yourself.")
        else:
            conn = sqlite3.connect('investigations.db')
            conn.execute("DELETE FROM users WHERE id=?", (user_id,))

```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

        now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        conn.execute("INSERT INTO audit_logs (username, action, timestamp)
VALUES (?, ?, ?)",
                    (session['username'], f"Deleted user ID {user_id}", now))

        conn.commit()
        conn.close()
        flash("User deleted.")
    return redirect(url_for('admin_panel'))

@app.route('/admin/logs')
def view_logs():
    if session.get('role') != 'admin': return redirect(url_for('home'))
    conn = sqlite3.connect('investigations.db')
    df = pd.read_sql("SELECT * FROM audit_logs ORDER BY id DESC", conn)
    logs = df.to_dict(orient='records')
    conn.close()
    return render_template('logs.html', logs=logs)

if __name__ == '__main__':
    app.run(debug=True)

```

HTML:

admin.html:

```

{% extends "base.html" %}
{% block content %}
<div style="margin-bottom: 20px;">
    <a href="/admin/logs" style="padding: 10px 15px; background: #4c1d95; color:
white; text-decoration: none; border-radius: 4px;">View Audit Logs</a>
</div>
<div class="results-box">
    <h2>User Management</h2>
    <div class="table-container">
        <table class="data-table">
            <thead>

<tr><th>ID</th><th>Username</th><th>Name</th><th>Email</th><th>Role</th><th>Status
</th><th>Actions</th></tr>
            </thead>
            <tbody>
                {% for u in users %}
                <tr>
                    <td>{{ u[0] }}</td><td>{{ u[1] }}</td><td>{{ u[2]
}}</td><td>{{ u[3] }}</td><td>{{ u[4] }}</td>
                    <td style="color: {% if u[5] == 'approved' %}green{% else
}%orange{% endif %}; font-weight:bold;">{{ u[5] }}</td>
                    <td>
                        {% if u[5] == 'pending' %}
                        <form action="{% url_for('approve_user', user_id=u[0]) %}"
method="POST" style="display:inline;">
                            <button type="submit" style="background:green;
color:white; border:none; padding:6px 12px; cursor:pointer; border-
radius:3px;">Approve</button>
                        </form>
                        {% endif %}
                    {% if session['user_id'] != u[0] %}

```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

```

        <form action="{{ url_for('delete_user', user_id=u[0]) }}"
method="POST" style="display:inline;">
        <button type="submit" class="btn-
delete">Delete</button>
    </form>
    {% endif %}
</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
{% endblock %}

```

base.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Blockchain Anomaly Detection</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <div class="nav-bar">
        <div><strong><a href="/" style="color: white; text-decoration:
none;">Blockchain Analyzer</a></strong></div>
        <div>
            {% if session.get('user_id') %}
                <span style="color: #d8b4fe; margin-right: 15px;">
                    Operator: {{ session['username'] }}
                    ({{ if session['role'] == 'admin' %}Privileged{% else
%}Standard{% endif %})
                </span>
                <a href="/dashboard">Database</a>
                {% if session['role'] == 'admin' %}
                    <a href="/admin">Admin Panel</a>
                {% endif %}
                <a href="/profile">Profile</a>
                <form action="{{ url_for('logout') }}" method="GET"
style="display: inline;">
                    <button type="submit" style="color: #fbcfe8;">Logout</button>
                </form>
            {% else %}
                <span style="color: #d8b4fe; margin-right: 15px;">Guest
Mode</span>
                <a href="/login">Login</a>
                <a href="/register">Register</a>
            {% endif %}
        </div>
    </div>

    <main class="main-container">
        <div class="message-box">
            {% with messages = get_flashed_messages() %}
            {% if messages %}
                {% for message in messages %}
                    <p class="flash-message">{{ message }}</p>
                {% endfor %}
            {% endif %}
            {% endwith %}
        </div>
    </main>

```

		БоднарашикА.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

```

        {% block content %}{% endblock %}
    </main>

    <script src="{% url_for('static', filename='script.js') %}"></script>
</body>
</html>

```

change_password.html:

```

{% extends "base.html" %}
{% block content %}
<div class="login-container">
    <h2>Change Password</h2>
    <form method="POST">
        <div class="form-group"><label>Current Password:</label><input
type="password" name="current_password" required></div>
        <div class="form-group">
            <label>New Password:</label><input type="password" name="new_password"
required>
            <small style="color: #666;">* Min 8 chars, 1 uppercase, 1 lowercase, 1
number.</small>
        </div>
        <div class="form-group"><label>Confirm New Password:</label><input
type="password" name="confirm_password" required></div>
        <button type="submit" id="scan-btn">Update Password</button>
    </form>
</div>
{% endblock %}

```

dashboard.html:

```

{% extends "base.html" %}
{% block content %}
<div class="results-box">
    <h2>Saved Anomalies (Total: {{ records|length }})</h2>
    <div class="table-container">
        <table class="data-table">
            <thead>
                <tr>
                    <th>Hash</th><th>Time</th><th>Inputs</th><th>Outputs</th>
                    <th>Input USD</th><th>Output USD</th><th>Fee USD</th><th>Fee
%</th><th>Model</th>
                    {% if session.get('role') == 'admin' %}<th>Action</th>{% endif
%}
                </tr>
            </thead>
            <tbody>
                {% for row in records %}
                <tr>
                    <td>{{ row['hash'] }}</td><td>{{ row['time'] }}</td>
                    <td>{{ row['input_count'] }}</td><td>{{ row['output_count']
}}</td>
                    <td>{{ "{:, .2f}".format(row['input_total_usd']) }}</td>
                    <td>{{ "{:, .2f}".format(row['output_total_usd']) }}</td>
                    <td>{{ "{:, .2f}".format(row['fee_usd']) }}</td>
                    <td>{{ "{:, .4f}".format(row['fee_percent']) }}</td>
                    <td>{{ row['scan_type'] }}</td>
                    {% if session.get('role') == 'admin' %}
                    <td><button type="button" class="btn-delete"
onclick="openDeleteModal('{{ row['hash'] }}')">Delete</button></td>
                    {% endif %}
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</div>

```

		БоднараишкА.О.			ККИКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

```

        </tr>
        {% endfor %}
    </tbody>
</table>
</div>
</div>
</div>

<div id="customDeleteModal" class="custom-modal">
    <div class="custom-modal-content">
        <h3>Confirm Deletion</h3>
        <p>Permanently remove this transaction? This cannot be undone.</p>
        <form id="modalDeleteForm" method="POST" action="">
            <button type="button" class="btn-cancel"
onclick="closeDeleteModal()">Cancel</button>
            <button type="submit" class="btn-delete">Yes, Delete</button>
        </form>
    </div>
</div>
</div>
{% endblock %}

```

index.html:

```

{% extends "base.html" %}
{% block content %}
<div class="upload-box">
    <form method="POST" enctype="multipart/form-data">
        <div class="form-group">
            <label>Upload Block Dataset (.tsv/.csv):</label>
            <input type="file" name="dataset" accept=".tsv,.csv,.txt" required>
        </div>
        <div class="form-group">
            <label>Select Threat Model:</label>
            <select name="scan_type">
                <option value="micro">Transit / Peeling Chains (Vol: $1k-$20k,
In/Out ≤ 2, Fee > $50)</option>
                <option value="whale">High-Volume Consolidation (Vol > $100k,
Outputs ≤ 5)</option>
            </select>
        </div>
        <button type="submit" id="scan-btn">Scan Network</button>
    </form>
</div>
{% if table %}
<div class="results-box">
    <h2>Found: {{ count }} anomalies</h2>
    {% if session.get('user_id') %}
        <form action="/save" method="POST" style="margin-bottom: 15px;">
            <button type="submit" id="save-btn">Export to Database</button>
        </form>
    {% else %}
        <p style="color: #666; font-style: italic;">Log in to save these results
to the database.</p>
    {% endif %}
    <div class="table-container">{{ table | safe }}</div>
</div>
{% endif %}
{% endblock %}

```

login.html:

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

```
{% extends "base.html" %}
{% block content %}
<div class="login-container">
  <h2>Operator Authentication</h2>
  <form method="POST">
    <div class="form-group"><label>Username:</label><input type="text"
name="username" required></div>
    <div class="form-group">
      <label>Password:</label>
      <div class="password-wrapper">
        <input type="password" id="password" name="password" required>
        <button type="button" id="toggle-password" class="toggle-
btn">Show</button>
      </div>
    </div>
    <button type="submit" id="scan-btn">Sign In</button>
  </form>
</div>
{% endblock %}
```

logs.html:

```
{% extends "base.html" %}
{% block content %}
<div class="results-box">
  <h2>System Audit Logs</h2>
  <div class="table-container">
    <table class="data-table">
      <thead><tr><th>Log ID</th><th>Timestamp</th><th>User</th><th>Action
Details</th></tr></thead>
      <tbody>
        {% for log in logs %}
          <tr><td>{{ log['id'] }}</td><td>{{ log['timestamp']
}}</td><td><strong>{{ log['username'] }}</strong></td><td>{{ log['action']
}}</td></tr>
        {% endfor %}
      </tbody>
    </table>
  </div>
</div>
{% endblock %}
```

profile.html:

```
{% extends "base.html" %}
{% block content %}
<div class="admin-section">
  <h3>Update Account Details</h3>
  <form method="POST">
    <div class="form-group"><label>Username:</label><input type="text"
name="username" value="{{ user[0] }}" required></div>
    <div class="form-group"><label>Full Name:</label><input type="text"
name="full_name" value="{{ user[1] }}" required></div>
    <div class="form-group"><label>Email:</label><input type="email"
name="email" value="{{ user[2] }}" required></div>
    <button type="submit" id="scan-btn">Save Changes</button>
  </form>
  <hr style="margin: 25px 0;">
  <h3>Security</h3>
  <a href="/change_password" style="display: inline-block; padding: 10px 15px;
background-color: #d97706; color: white; text-decoration: none; border-radius:
```

		Боднарашик А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

```
4px;">Change Password</a>
</div>
{% endblock %}
```

register.html:

```
{% extends "base.html" %}
{% block content %}
<div class="login-container">
  <h2>Account Registration</h2>
  <form method="POST">
    <div class="form-group"><label>Full Name:</label><input type="text"
name="full_name" required></div>
    <div class="form-group"><label>Email:</label><input type="email"
name="email" required></div>
    <div class="form-group"><label>Username:</label><input type="text"
name="username" required></div>
    <div class="form-group">
      <label>Password:</label><input type="password" name="password"
required>
      <small style="color: #666;">* Min 8 chars, 1 uppercase, 1 lowercase, 1
number.</small>
    </div>
    <button type="submit" id="scan-btn">Register</button>
  </form>
</div>
{% endblock %}
```

CSS:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f8f6fa;
  color: #1e1b4b;
  margin: 0; padding: 0;
}

.nav-bar {
  background-color: #1f112e;
  padding: 15px 20px;
  color: white;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.nav-bar a, .nav-bar button {
  color: #e9d5ff;
  text-decoration: none;
  margin-left: 15px;
  background: none;
  border: none;
  cursor: pointer;
  font-size: 15px;
}

.nav-bar a:hover, .nav-bar button:hover {
  color: #ffffff;
  text-decoration: underline;
}
```

		Боднаришук А. О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

.main-container {
  max-width: 1200px;
  margin: 30px auto;
  padding: 0 20px;
}

.admin-section, .login-container, .results-box, .upload-box {
  background-color: #ffffff;
  border: 1px solid #d8b4fe;
  border-radius: 6px;
  padding: 25px;
  margin-bottom: 25px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.05);
}

.login-container {
  max-width: 400px;
  margin: 60px auto;
}

.form-group { margin-bottom: 18px; }
.form-group label { display: block; font-weight: bold; margin-bottom: 6px; color: #4c1d95; }
.form-group input, .form-group select {
  width: 100%; padding: 10px;
  border: 1px solid #c084fc; border-radius: 4px; box-sizing: border-box;
}

.password-wrapper { display: flex; }
.password-wrapper input { border-radius: 4px 0 0 4px; border-right: none; }
.toggle-btn {
  background-color: #f3e8ff; border: 1px solid #c084fc;
  padding: 0 15px; border-radius: 0 4px 4px 0;
  cursor: pointer; color: #4c1d95; font-weight: bold;
}

#scan-btn, #save-btn {
  background-color: #5b21b6; /* Темно-фіолетовий */
  color: white; padding: 10px 20px; border: none; border-radius: 4px;
  cursor: pointer; font-weight: bold; width: 100%;
}

#scan-btn:hover, #save-btn:hover { background-color: #4c1d95; }

.btn-delete {
  background-color: #be185d; /* Рубіновий */
  color: white; border: none; padding: 6px 12px;
  cursor: pointer; border-radius: 3px; font-weight: bold;
}

.btn-delete:hover { background-color: #9f1239; }

.btn-cancel {
  background-color: #6b7280; color: white; border: none;
  padding: 6px 12px; cursor: pointer; border-radius: 3px; margin-right: 10px;
}

.table-container {
  overflow-x: auto;
  border: 1px solid #d8b4fe;
  border-radius: 4px;
}

.data-table {
  width: 100%; border-collapse: collapse; white-space: nowrap;
}

```

		Боднарашиш А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

```

.data-table th {
  background-color: #f3e8ff; color: #4c1d95;
  padding: 12px; text-align: left; border-bottom: 2px solid #c084fc;
}

.data-table td {
  padding: 10px 12px; border-bottom: 1px solid #e9d5ff; color: #1e1b4b;
}

.data-table tr:hover { background-color: #faf5ff; }

.flash-message {
  background-color: #fce7f3; border-left: 4px solid #be185d;
  color: #831843; padding: 10px 15px; margin-bottom: 15px;
}

.custom-modal {
  display: none; position: fixed; top: 0; left: 0; width: 100%; height: 100%;
  background-color: rgba(15, 10, 25, 0.7); justify-content: center; align-items:
center; z-index: 1000;
}

.custom-modal-content {
  background-color: white; padding: 30px; border-radius: 8px;
  text-align: center; width: 400px; box-shadow: 0 4px 15px rgba(0,0,0,0.3);
}

.custom-modal-content h3 { color: #9f1239; margin-top: 0; }

```

JavaScript:

```

document.addEventListener('DOMContentLoaded', function() {
  const headers = document.querySelectorAll('th');
  if (headers.length > 0) {
    headers.forEach(header => {
      header.style.cursor = 'pointer';
      header.addEventListener('click', () => {
        const table = header.closest('table');
        const tbody = table.querySelector('tbody') || table;
        const rows = Array.from(tbody.querySelectorAll('tr'));
        const index =
Array.from(header.parentNode.children).indexOf(header);

        const isAscending = header.classList.contains('asc');
        headers.forEach(h => h.classList.remove('asc', 'desc'));
        header.classList.add(isAscending ? 'desc' : 'asc');

        rows.sort((a, b) => {
          const aText = a.children[index].innerText;
          const bText = b.children[index].innerText;
          const aNum = parseFloat(aText.replace(/,/g, ''));
          const bNum = parseFloat(bText.replace(/,/g, ''));

          if (!isNaN(aNum) && !isNaN(bNum)) return isAscending ? bNum -
aNum : aNum - bNum;
          return isAscending ? bText.localeCompare(aText) :
aText.localeCompare(bText);
        });
        rows.forEach(row => tbody.appendChild(row));
      });
    });
  }

  const togglePasswordBtn = document.getElementById('toggle-password');
  const passwordInput = document.getElementById('password');

```

		Боднарашик А.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    if (togglePasswordBtn && passwordInput) {
        togglePasswordBtn.addEventListener('click', function () {
            const isPassword = passwordInput.getAttribute('type') === 'password';
            passwordInput.setAttribute('type', isPassword ? 'text' : 'password');
            this.textContent = isPassword ? 'Hide' : 'Show';
        });
    }
});

window.openDeleteModal = function(txHash) {
    document.getElementById('modalDeleteForm').action = '/delete/' + txHash;
    document.getElementById('customDeleteModal').style.display = 'flex';
};

window.closeDeleteModal = function() {
    document.getElementById('customDeleteModal').style.display = 'none';
};

```

.env:

```
MY_SECRET_KEY="HashSaltEnc123!"
```

		БоднарашикА.О.			ККІКБ.КР.Б – 125 – 02 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		67